(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 August 2001 (23.08.2001)

PCT

(10) International Publication Number
WO 01/61508 A1

(51) International Patent Classification[7]: G06F 13/00, 15/16, H04L 9/00

(21) International Application Number: PCT/US01/04812

(22) International Filing Date: 14 February 2001 (14.02.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/183,681	19 February 2000 (19.02.2000)	US
60/191,778	24 March 2000 (24.03.2000)	US
09/636,102	10 August 2000 (10.08.2000)	US

(71) Applicant *(for all designated States except US)*: **DIGIMARC CORPORATION** [US/US]; 19801 SW 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(72) Inventors; and

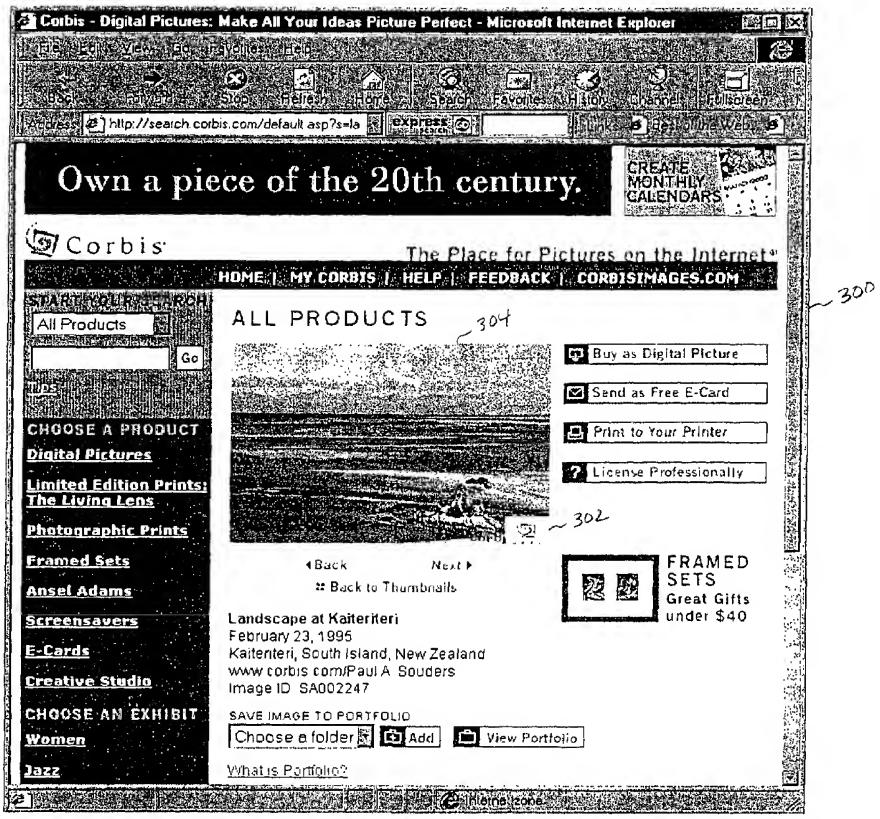
(75) Inventors/Applicants *(for US only)*: **RAMOS, Daniel, O.** [US/US]; 16869 SW Hargis Road, Beaverton, OR 97007 (US). **JONES, Kevin, C.** [US/US]; 4850 NW Neskowin Ave., Portland, OR 97229 (US). **RHOADS, Geoffrey, B.** [US/US]; 2961 SW Turner Road, West Linn, OR 97068 (US).

(74) Agent: **MEYER, Joel, R.**; Digimarc Corporation, 19801 S.W. 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: WATERMARK ENCODER AND DECODER ENABLED SOFTWARE AND DEVICES



(57) Abstract: Watermark encoders and decoders are integrated into operating systems, Internet browsers (300), media players, and other applications and devices. Such integration enables the watermark-enabled application (304) or device to provide additional functionality and information (302) available via the watermark. The watermark, for example, may link to metadata or actions related to a media object. To exploit this watermark enabled functionality, the integrated application uses a watermark decoder to access the related metadata and actions. The user interface of the integrated application is enhanced to present metadata and actions linked via the watermark. Similarly, watermark encoders may be integrated into applications to convert media objects into enhanced, watermarked objects.

WO 01/61508 A1

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

WATERMARK ENCODER AND DECODER ENABLED SOFTWARE AND DEVICES

Related Application Data

This patent application claims priority to US Provisional Application No. 60/191,778, filed March 24, 2000, which is hereby incorporated by reference. This application also is a continuation in part of applications 09/165,142 filed October 1, 1998, 09/503,881, filed February 14, 2000, 09/507,096, filed February 17, 2000, 09/526,982, filed March 15, 2000, 09/531,076, filed March 20, 2000, and 09/620,019, filed July 20, 2000, which are hereby incorporated by reference.

This patent application is also related to US Patent Applications 09/525,865 entitled Integrating Digital Watermarks into Multimedia Content filed March 15, 2000, 09/563,664 entitled Connected Audio and Other Media Objects filed May 2, 2000, 09/571,422 entitled Methods and Systems for Controlling Computers or Linking to Internet Resources from Physical and Electronic Objects filed May 15, 2000, and 09/574,726, entitled Methods and Systems Employing Digital Watermarking filed May 18, 2000, which are hereby incorporated by reference.

Technical Field

The invention relates to digital watermarking, and specifically relates to applications of digital watermark encoders and decoders in software and devices.

Background and Summary

Digital watermarking is a process for modifying physical or electronic media to embed a machine-readable code into the media. The media may be modified such that the embedded code is imperceptible or nearly imperceptible to the user, yet may be detected through an automated detection process. Most commonly, digital watermarking is applied to media signals such as images, audio signals, and video signals. However, it may also be applied to other types of media objects, including documents (e.g., through line, word or character shifting), software, multi-dimensional graphics models, and surface textures of objects.

Digital watermarking systems typically have two primary components: an encoder that embeds the watermark in a host media signal, and a decoder that detects and reads the embedded watermark from a signal suspected of containing a watermark (a suspect signal). The encoder embeds a watermark by altering the host media signal. The reading component analyzes a suspect signal to detect whether a watermark is present. In applications where the watermark encodes information, the reader extracts this information from the detected watermark.

A great number of particular watermarking techniques are known. The reader is presumed to be familiar with the literature in this field. Particular techniques for embedding and detecting imperceptible watermarks in media signals are detailed in the present assignee's copending application serial number 09/503,881. Other watermarking techniques are known from published patents to NEC (inventor Cox et al), IBM (inventors Morimoto and Braudaway et al), Dice (inventor Cooperman), Philips (inventors Kalker, Linnartz, Talstra, etc. Audio watermarking techniques are known from published patents to Aris (inventor Winograd, Metois, Wolosewicz, etc.), Solana (inventor Lee, Warren, etc.), Dice, AudioTrack, Philips, etc.

This invention relates to integrating watermark encoding and decoding functions in software applications, devices and systems. Watermark encoders and decoders are integrated into operating systems, Internet browsers, media players, and other applications and devices. Such integration enables the watermark-enabled application or device to provide additional functionality and information available via the watermark. The watermark, for example, may link to metadata or actions related to a media object. To exploit this watermark enabled functionality, the integrated application uses a watermark decoder to access the related metadata and actions. The user interface of the integrated application is enhanced to present metadata and actions linked via the watermark. Similarly, watermark encoders may be integrated into applications to convert media objects into enhanced, watermarked objects.

One aspect of the invention is an enhanced file browser system. The file browser enables the user to browse files, including media object files. It includes an extension to decode an object identifier from a selected media object file. The

extension retrieves and displays metadata or actions associated with the media object file via the object identifier.

Another aspect of the invention is a file browser with an extension for encoding an object identifier. This extension enables the user to encode an object identifier into a selected media object file. In an extension to the file browser's user interface, it displays options for controlling the encoding of the object identifier. The object identifier may be encoded into a watermark embedded in the media object.

Another aspect of the invention is a watermark decoder system incorporated into a host application. The host application has a user interface for displaying a representation of media object files. It also has an extension for decoding a watermark from a selected media object file and for displaying metadata associated with the media object file via the watermark.

Another aspect of the invention is an enhanced internet browser. The browser is enhanced with a listener program that identifies media objects in an HTML document, and inserts a handler into the HTML document when it finds an object marked with an object identifier. The handler displays metadata linked via the object identifier in response to user input.

Another aspect of the invention is a method of rendering a media object with branding information. The method decodes an object identifier from the media object, sends the object identifier to a metadata server, receives a brand identifier from the metadata server, and displays a representation of the brand identifier.

Another aspect of the invention is a method for extending a user interface of a media player. In response to input requesting playback of a media object, the method extracts an object identifier from the media object. It uses the object identifier to look up metadata associated with the media object. It then extends the user interface of the media player to include a representation of the metadata associated with the media object.

Further features of the invention will become apparent with reference to the following detailed description and accompanying drawings.

Brief Description of Drawings

Fig. 1 is an example of user interface features enabled by integrating a watermark decoder in an operating system or other application program.

Fig. 2 is an alternative implementation of a user interface for displaying metadata in the file browser of Fig. 1.

Fig. 3 is an example of user interface features enabled by integrating a watermark decoder in an Internet browser.

Fig. 4 shows the example of Fig. 3 with an expanded menu of options linked to a media object via a watermark.

Fig. 5 is an example of user interface features enabled by integrating a watermark decoder in a media player.

Fig. 6 is an example of a user interface features enabled by integrating a watermark encoder in an operating system or other application program.

Fig. 7 is a diagram of a computer system that serves as an operating environment for software implementations of watermark encoder/decoder enabled applications.

Fig. 8 is a diagram with an image containing the words "Confidential".

Fig. 9 is diagram of the fields in a typical digital watermark.

Fig. 10 is a diagram of a typical e-mail system.

Fig. 11 is a more detailed diagram of the watermark reading and detection program shown in Fig. 10.

Fig. 12 is a diagram illustrating a content filtering and indexing system with watermark content filters.

Fig. 13 is a diagram illustrating a distributed watermark detector system.

Detailed Description

Introduction

A watermark may be used to associate a media object such as an image, video or audio file to additional information and actions. The watermark can associate the media object with metadata or processing actions encoded within or stored outside the

media object. Metadata may be encoded in a watermark message, stored within the media object file, or stored outside the media object file. When metadata is stored outside the media object, the watermark may encode an imperceptible and persistent link to this metadata, such as an object identifier (number, address, index, pointer, etc.) within the media object. Wherever the media object travels, watermark decoder-enabled software or devices can extract the watermark from the media signal, and access the metadata or actions associated with the object via the watermark link.

The specific infrastructure for retrieving metadata or actions associated with a media object via its watermark may vary. The metadata or processing actions may reside in a metadata database in the same device or system as the media object, or in a remote device or system accessible via a wire or wireless network. In a distributed computing environment like the Internet, one way is to implement a database server that takes a object identifier extracted from a watermark message and performs one or more tasks associated with the identifier, such as returning metadata or URL links to a requesting computer or device ("client"), routing the identifier to another server, executing some program or set of programs, etc. The watermark message refers to the auxiliary data encoded into a host media object via the watermark.

The following sections summarize ways to take advantage of this functionality in an operating system, Internet browser, and other applications (whether implemented in hardware devices, software, or a combination of hardware and software).

Integrating a Watermark Decoder in Operating Systems and Other Applications

In Operating System

A watermark decoder may be integrated in a file browser component of an operating system. The decoder enables the file browser to extract a watermark and retrieve metadata for watermark enabled media objects. As the user browses files, the watermark decoder may operate as a foreground or background task, automatically or user-initiated, to detect a watermark. Finding a watermark, the browser annotates its representation of the media object as directed by the application(s) associated with the

watermark. Depending on the implementation, the browser may proceed to retrieve additional information or actions associated with the object via the watermark and annotate its representation of the media object accordingly. For example, the browser may annotate the media object with a description of information or actions linked via the watermark (e.g., a brief description and/or http link), or may annotate the object with the actual information or actions.

The process of detecting a watermark and referencing information or actions via the watermark may be implemented to be transparent to the user. For example, the file browser displays information or options obtained via the watermark link without requiring the user to intervene in the watermark detection or information retrieval process.

Alternatively, the file browser can give the user the opportunity to control various stages of watermark detection and processing triggered by the watermark payload. For instance, the user may be given the option to allow the watermark decoder to operate on media objects, and to determine whether and how actions triggered by the watermark payload should proceed. Upon detection of a watermark, for example, the media object can be annotated with an indicator, such as a distinctive sound or logo, that informs the user that information and actions can be accessed via a link embedded in the watermark. The user has the option to access additional information associated with the media object by, for example, selecting a visual logo associated with the object in the user interface. An audio "logo" may be played when the user selects the object (e.g., passes a cursor over its graphical representation in the user interface).

The user interface of the application can be annotated with a variety of graphical and/or audio effects that inform the user of the presence of the watermark link and associated information and actions. Changes in the user interface may be used to convey different stages in the watermark detection and metadata retrieval process. For example, when it first detects the presence of a watermark, the decoder (or host application) plays a generic indicator, such as a simple logo or audio clip. Then, when the appropriate metadata server returns metadata and/or instructions linked via the watermark, the user interface presents specific information associated with the object.

The server may return program code, such as Java Applets, Visual Basic script, XML or some other set of instructions, that present information to the user and provide links to additional information and actions (e.g., URLs or hot links to web sites, other content or program code). Upon receiving this code, a client computer or device executes it. The client typically is the computer or device that decoded the watermark link and issued a request based on the link to the server. This code may perform a variety of functions, including controlling rendering of the watermarked media object and related media objects, some of which may be returned with the linked metadata. The server may return code to control decoding or decrypting the media object or other related media signals to be played along with the watermarked media object. In addition, the server may return code and/or links to enable the user to establish a license and obtain usage rights electronically with a licensing program executing locally and/or on a remote licensing computer (e.g., a licensing server on the Internet).

The server may return data, such as XML, that defines actions to be taken (by providing URLs, instructions, etc.). The client computer or device receiving such a definition of actions may execute the action or present them to the user as options to be executed in response to user input (e.g., clicking on a graphical representation of the option in the user interface of a computer, responding to voice commands via a speech recognition engine, etc.).

The user can access linked information or actions by selecting a graphical representation of the object in a user interface. For example, the user can "click-on" an icon representing the object, or a rendered version of the object (e.g., an image) to determine whether metadata or actions are linked to the object, or to initiate a linked action or retrieval of the metadata.

The watermark decoder may be designed to search for the presence of watermarks in media objects in a specified location (e.g., directories, hard drive, etc.) in response to an event, at periodic intervals, or in response to a user request. For instance, the watermark decoder may be implemented as a utility service, similar to a file search utility, that the user may invoke to a extract watermark link from a media object file, or from a group of files. As another example, the operating system can be designed to invoke the decoder at boot up on all files of a given type in a selected

storage location (e.g., on the hard drive). The operating system may also run the decoder as a background utility to periodically check for watermark links in media objects. A timer or clock service may be used to trigger watermark detection when the timer elapses or at pre-determined time intervals. The operating system may also run the decoder when prescribed events happen, such as downloading a file, saving a file, etc. Triggering the decoder in response to such events enables the operating system to ensure that media objects are checked whenever they enter the system or device in which the operating system is executing, and whenever they are edited.

To improve the efficiency of the watermark decoder, the file system may implement a scheme for tracking when and which media objects have been checked for watermarks. One such scheme is to assign attributes to each media object to indicate whether it has been checked, whether or not it is watermarked, and if so, when the watermark was detected. Then, the decoder uses this information to check only media objects that have not been checked or have been modified, or for media objects for which a given period of time has elapsed since the last check. Each time a media object is modified, the attribute indicating that the mark has been checked may be reset to ensure that only new and modified objects are re-checked.

To illustrate the concept, consider an implementation in the Windows Operating System. Fig. 1 illustrates an example of an extension of the Windows Explorer user interface to support watermark embedding and reading of media objects. Media object files are typically represented as icons 100 in the user interface 102 of the Windows Explorer file browser. By right clicking the mouse while positioning a cursor over the file icon 100, the user can access a context menu 104 of options associated with the selected media object.

This "watermark aware" file browser augments the options of the context menu by listing options such as "Read Watermark" or "Watermark." When the user positions the cursor over the "Read Watermark" option, the operating system invokes a watermark decoder on the media object. The watermark decoder extracts the watermark link and acts in concert with network communication software to access the metadata database and retrieve the items from the database associated with the watermark link. The file browser displays these items in a window 106.

There are a variety of ways to access the metadata database. The metadata may be stored locally (in the same machine as the media object), in a local area network or a wide area network. If the database is located on a remote computer on a computer network, such as the Internet, network communication software may be used to establish a connection with the database.

When selected, the "Watermark" option displays further options for reading and embedding information ("Read Information..." and "Embed Information..."). The window labeled Image Information 106 displays metadata and actions associated with an image via a watermark link embedded in it. In this example, the window 106 displays a thumbnail of the image, image attributes (e.g., width, height, resolution, bits per pixel), and a series of HTML links 108-116 to additional information and actions (such as a search for related images at links 114, 116).

Another way to display items linked via a watermark is to insert them in an additional property page as shown in Fig. 2. While executing the file browser, the user accesses properties (option 118 in Fig. 1) by right clicking the mouse while positioning the cursor over the media object's icon. In response to selecting properties, the operating system displays a properties window such as the one shown in Fig. 2. Each of the property pages associated with the media object has a tab (e.g., General, Image, Watermark, Security). The Watermark page is selected in Fig. 2. Like the options displayed in the window 106 of Fig. 1, the Watermark page lists a series of HTML links 202-210 to additional information or actions. When selected, the link invokes an Internet browser to retrieve information at the underlying URL. For actions like searches 208-210, the link may pass additional parameters such as attributes of the media object to the server at the URL, which in turn, executes a search using those parameters and returns the results to the Internet browser.

An additional way to reflect that a file includes watermarked data is to superimpose a graphical "watermark indicator" icon to the file's icon to signify the presence of a watermark in the file.

While the implementation may vary, the examples depicted in Figs. 1 and 2 are shell extensions of the Windows Explorer file browser. The decoder is implemented as a COM object client of the Windows Explorer. The COM object client is registered

with the operating system as a shell extension handler. Fig. 1 depicts a context menu handler, while Fig. 2 depicts a properties page handler.

Another possible implementation is to implement a shell extension that uses a shell execute command to launch a metadata retrieval application that gets and displays metadata options. This implementation adds an extension to the file browser user interface, such as a context menu extension. When the user selects a media file object within the file browser user interface, it displays a context menu with an option to launch the metadata retrieval program associated with media objects of a given type. A number of actions can be tied to this option. One action is to launch the metadata retrieval application program. Another action is to launch a media player to play the selected option. Of course, both actions can be initiated concurrently in a multitasking operating system.

One example of a metadata retrieval application is a watermark decoder that extracts a watermark message, and forwards an object identifier from the message to a metadata server, which then returns metadata. As noted above, the retrieval application need not extract the object identifier from a watermark if it was already extracted and used recently to retrieve metadata. Instead, the retrieval application can proceed to display the metadata and actions to the user.

To launch the retrieval application, the shell execute command passes the name and location of the media object to the retrieval application. The retrieval application may present its own user interface to display linked metadata and actions, or may pass them to the file browser, which then displays them within an extension such as a properties page or context menu extension. The retrieval application may prompt the user to request permission before decoding a watermark or requesting an update of metadata from the metadata server. Additionally, the retrieval application may launch one or more other applications, such as an Internet browser to issue a request for metadata from a Web server and display metadata and actions in an HTML document returned from the server.

The approaches described above can be implemented for a variety of media object files, including image, video and audio files. Also, the object identifier need not

be inserted in a watermark, but instead may be placed somewhere else in the media object file, such as a file header.

In Browser

The watermark decoder may also be integrated into an Internet browser application. Like the file browser, an Internet browser can browse directories of files in a local computer or across a network. Commercially available browsers like Internet Explorer from Microsoft Corporation and Netscape Navigator typically support transfer protocols like HTTP and FTP, and have additional components for interpreting or compiling code, including HTML, Java, Visual Basic and scripts (e.g., Java scripts, Visual Basic scripts, etc.). In addition, Internet browsers can parse and render HTML for display in a computer or other device's user interface.

To illustrate integrating a watermark decoder in an Internet browser, consider the following example. As the Internet browser downloads and parses web pages with media objects on the Internet, it keeps a listing of these objects, and checks them for the presence of a watermark. In the listing, it annotates the representation of the objects with watermarks to reflect the presence of the watermark, and potentially other data such as a URL where the media object originated. The user may access the listing by viewing an application window (e.g., an application bar) that presents a visual representation of the media objects. For images, one way to represent the media object is through the use of a thumbnail of the image. Images and other objects may be represented as a graphical icon, textual description, or both.

Watermark objects may be distinguished with a visual or audio indicator like a distinctive sound or logo. The user views the metadata or actions associated with a media object by selecting the representation of the watermark-enabled object in some fashion. For example, the user can click on the thumbnail, icon, logo, or textual description to access a menu of metadata or actions linked to the object via the watermark.

Another way to indicate watermarked objects is to alter the appearance of a cursor in a graphical user interface of the software application when the user passes the cursor over the watermarked object or a representation of the object displayed in the

graphical user interface. One way to alter the cursor is to change it from a conventional pointer to a distinctive graphical icon associated with the detected watermarked object type. Another way is to animate the cursor such that it morphs into some other shape or graphical design over a sequence of frames. Similarly, the user interface can alter the appearance of the watermarked object as the user passes the cursor over it. In addition, the user interface can produce a distinctive sound when the user passes the cursor over the object to signify that it includes a watermark.

US Patent Application No. 09/165,142 provides an example of how to decode watermarks from images in HTML pages and annotate the images with a logo indicating that a watermark is present. It describes a way to present a representation of media objects (thumbnails of images) in an application window of a browser. One such application window may be used to display thumbnails of images to present a history of images encountered while browsing web pages on the Internet or elsewhere. The user may click on an image to add it to a separate "bookmark" or "favorites" list. Another application window may be used to display thumbnails of the images in this bookmark list. By selecting an image or a representation of it, the browser links to a network resource associated with the selected image (e.g., via an associated URL). This network resource may be a web page where the image originated. Alternatively, the resource may be a web page referenced via the watermark link embedded in the image. The watermark message may encode a URL or an object identifier that is used to look up a URL of a network resource, such as a web page. For example, the URL might link to the web page of an owner, or to a licensing server.

The methods described in US Patent Application No. 09/165,142 may be applied to other media objects like video and audio signals.

One way to allow the user to access metadata and actions linked to a media object via a watermark is to display them in a menu in the user interface of the Internet browser. Fig. 3 shows an example of how to display metadata and actions associated with a media object in the user interface 300 of an Internet browser. A browser listener program receives events from the Internet browser indicating when a web page has been downloaded. The listener requests from the browser the address or addresses of media objects in the web page. The address indicates where the media object resides

in memory (main memory, virtual memory, cache, etc.). The listener invokes a watermark decoder on the media object or objects, passing it the address of the object in memory.

Finding a watermark, the listener inserts a handler program into the web page in memory. This handler program is responsible for presenting a logo or other indicator to the user indicating the presence of the watermark and providing hot links to information and actions. For example, the indicator in Fig. 3 is a logo 302 superimposed over a rendered version 304 of a watermarked image object in the browser's user interface.

When the user passes the cursor over a logo and selects it, the handler program associated with it displays a menu 400 of options as shown in Fig. 4. The listener program retrieves these options by establishing a connection to a metadata server (e.g., a local or remote database), passing an object identifier extracted from the watermark to the server, and receiving associated items from the server. These items may include URL links to web pages related to the media object, information about the media object, or links to an action, such as licensing server.

While specific implementations may vary, the example depicted in Figs. 3 and 4 is implemented using document view extensions to the Internet Explorer browser from Microsoft Corporation. Microsoft's dynamic HTML provides an interface that allows an Internet Explorer listener program to insert code to modify an HTML document. Using this interface, the listener program inserts a Java script that controls the display and responds to input to the logo superimposed on the image shown in Fig. 3.

In some applications, media content and web site developers may wish to selectively enable or disable functionality associated with a watermark link. One way to implement this feature is to modify the media object file or some other file that acts as a container of a media object (e.g., an HTML) to include a control parameter (like a flag in a header file). This control parameter indicates the presence of a watermark enabled media object and whether the watermark link is enabled or disabled. The control parameter may be designed to default to being active, unless expressly turned off, or vice-versa. When content developers include the object in some multimedia work, such as a website, they can opt to disable the watermark link via the control

parameter. Editing tools and other application programs and devices can be designed to turn the flag on unless expressly instructed to disable the watermark link.

In other applications, it is important that the watermark act as a persistent link to associated metadata. As the media object travels through different systems, gets coded/decoded, gets modified, etc., it may lose conventional metadata stored in the media object file. In these cases, the watermark link may be used to restore the metadata because the watermark is robust to various forms of transformations (e.g., digital to analog conversion, analog to digital conversion, compression, etc.).

Another way to selectively control functionality made available via the watermark is to enable the user to enable or disable watermark decoding on media objects.

The watermark can also be used as an attribute to the browser's file cache system. This allows cache browser applications to identify which cached files are watermarked and perform functions associated with the information embedded in the watermark as described throughout this document.

In Other Applications

The features outlined above can be implemented in any software applications or devices that process media objects. For example, a media object database management system may implement similar functionality. Digital asset management and digital rights management systems may use watermark enabled links to track and control the use of media objects.

Other applications that encounter media objects, like Word processing, spreadsheet, presentation programs, media object editing tools, etc. can all use some version of the features outlined above to link media objects with additional information and actions.

Many application programs have file browser capabilities that can be enhanced using the technology described in this document. For example, the File Open command is often used to browse file objects. The context menu and properties page extensions described above can be implemented for applications that provide file browsing services.

Watermark encoding and decoding functions can also be integrated into file sharing systems, such as the peer to peer file sharing systems like Napster, Gnutella, Freenet, Scour, etc. In a file sharing system, file sharing software executes on a number of client computers interconnected on a computer network. This software tracks the files available for sharing in the computer in which it executes, as well as other computers interconnected via the network. File sharing software may use watermarks or other forms of embedded data in files to control file transfers (uploading and downloading files on the computer), verify that a file is complete and free of viruses, carry metadata that may be used to search for files in the file sharing system, carry links to additional information and opportunities to obtain usage rights or to buy intellectual property rights in the file, or related products or services. To access this functionality, the file sharing software includes watermark or other embedded data decoding software. To insert or alter this functionality, the file sharing software includes watermark or other embedded data encoding software. For more information on this application, see US Patent Application No. 09/620,019, filed July 20, 2000, and entitled Using Embedded Data with File Sharing which is incorporated by reference above.

Context Sensitive Watermark Links

The actions or information linked to a media object via a watermark may be context sensitive. Some examples of "context" include application context (e.g., referring to the application program that is operating on the object), the object location context (where the object resides relative to the user). For example, the behavior of the link may be different when the user is manipulating the object in an editing tool as opposed to inserting the object in a document, such as a word processing document, a spreadsheet, or presentation. Also, the behavior of the watermark link may change based on whether the object is local, in an intranet, or on the Internet, relative to the user.

The decoder application may link to different metadata or processing actions depending on the context of the media object. To implement this functionality, the watermark decoder provides context information to the metadata database, which in

turn, provides context specific metadata or initiates context specific actions. For instance, if the object is being edited with an editing tool, the decoder provides information about the editing tool to the database in addition to the link extracted from the watermark. Similarly, the decoder may provide context information indicating where the object resides relative to the user's computer. Using the context information as a key, the metadata database returns metadata or initiates processing actions that are associated with the context.

Another way to implement the context sensitive behavior is to allow the decoder to control the presentation of watermark-linked actions or information based on the context of the media object. In the case where the context is defined by the application, this approach is akin to giving the application access to all of the linked metadata or actions, and then letting the application control presentation of the linked actions or information to the user. In such a scenario, the metadata database returns at least a descriptor of linked information and actions, and the application chooses which sub-set of the information or actions to apply based on the context. In the case where the context is defined by the location of the object, the decoder operates in a similar fashion. For example, it may choose a sub-set of the linked actions of information based on the location of the object.

A specific example of context information is user information supplied by the user's computer to a web server over the Internet using "cookie" technology. First, the user's computer decodes a watermark from a media object, such as an image, audio, or video signal. It then sends information extracted from the watermark along with a cookie including user information to a metadata server or router (generally referred to as a server). The server operates on user information from a cookie along with information extracted from a watermark in a media object to look up information or actions that are personalized to the user. The metadata server, for example, parses the cookie and uses it to reference information or addresses to network resources (URLs of web pages) in a database that relate to information in the cookie. The server further narrows the pertinent information or links by using the watermark information to look up a subset of information or links that are pertinent to the watermarked object. Then either the server, or another network resource referenced by the database operations

returns associated information back to the users computer for rendering on the display or audio output device. This approach can be used to limit the information returned to specific news, advertising, content, etc. that is likely to be of interest to the user. A similar effect can be achieved by programming the user's computer to supply user preferences that are used in a similar manner as the cookie information.

Supporting Multiple Watermark Types

As watermark technology proliferates, media objects may have different types of watermarks, each associated with a set of watermark encoders and decoders. To accommodate different watermark types, the decoder can be designed to support different watermark protocols. The watermark protocols provide keys and other parameters specifying how to decode a watermark of a given type. Alternatively, a common Application Programming Interface (API) can be specified for different core watermark encoder and decoder software modules or devices. These schemes facilitate the development of many different types of applications and devices that invoke watermark encoder and decoder functions, yet are independent of the watermark protocol and/or core watermark methods.

To support different core watermark methods, the user may install two or more different core watermark encoder/decoder modules. For example, the core modules may be implemented as plug-ins or dynamic link libraries. When installing a module, the installation process updates a registry, such as the registry in the Windows Operating System, to reflect that a watermark type is supported. In this manner, watermark decoders for different media types, and different types of decoders for a single media type may be supported.

In cases where a media object contains a watermark of unknown type, the media object file may specify the watermark type, e.g., through a parameter in a file header. The file browser, or other client of the core watermark module, may invoke the appropriate decoder by extracting the type parameter from the media object and passing it and a reference to the media object to the core module via the API. The API routes the request to the appropriate core module, which in turn, extracts the watermark

message, and returns it to the API. The API passes the message to the requesting application.

In the event that a type parameter is not available, the application or device processing the object may enumerate through all supported watermarking protocols to check if any protocol is present. The watermark protocols for given media or file type may be registered in the device or application (e.g., in a registry of the operating system). To check for each one, the application invokes a watermark screening process for these protocols to determine whether a watermark associated with the protocols is present.

Media Object Branding

The watermark can be used to establish "branding" in addition to facilitating electronic access to other services. In such a branding application, a watermark decoder enabled application or device in a client reads the embedded watermark message from the object and use it to access a digital logo, such as a thumbnail image (e.g., a brand "brand image"). The decoder-enabled application sends an object identifier taken from the watermark message to a server (such as a metadata server on the Internet), which, in response, returns the logo. The application then renders the logo, preferably superimposed on a rendered version of the media object or a graphical representation of it in the user interface of a client computer or device.

In some scenarios, the server also returns an address (e.g., URL) pointing to either a generic "usage rights" server or a custom "usage rights" maintained by the media owner. In an Internet context, the server may return one or more links to related Web sites. Connection rights and corporate branding services may be provided via a central server on the Internet.

Using the object identifier to link to actions and metadata, watermark enabled applications have many options to extend the branding and usage rights services. An application could display "updatable" usage rights associated with the object identifier. In visual media objects, the application may display the branding logo visually superimposed over a portion of the rendered object (e.g., in a corner of a video frame, periodically for a predetermined period of time) or displayed as a splash screen

initiation before playback of the media object begins. To make the branding information less obtrusive, it may be accessible upon request through a menu (e.g., when the user clicks on a representation of the object or "help" menu).

The branding service transforms the notion of copyright notification into a substantial ever-present branding opportunity with additional functionality, such as a hot link to the home page of the content owner or to a licensing server. The branding service may be combined with other watermark enabled functionality, such as a copy management instruction in the watermark payload, e.g., a control parameter indicating whether the object may be played, copied (number of copies allowed), recorded, transferred into other device or system, etc. In addition to providing this instruction to control usage, the watermark payload provides additional value to the consumer (e.g., linking to additional information and services associated with a media object) and ensures that the media object is well labeled and branded during playback.

More elaborate hot branding links, usage rights services, and context-sensitive linking may be added by associating the watermark link with software programs, metadata, and pointers to programs and metadata that support these features. These features may be added in the metadata server by adding them to the list of actions and/or metadata to be executed or returned in response to receiving the media object identifier. In addition, the decoder enabled application may be programmed to send the media object identifier to two or more servers that provide different sets of services or metadata for the object identifier.

Aggregating Metadata

In some applications, a media object may be associated with metadata from two or more different sources. The metadata may be stored in a file that stores the media object. For instance, file formats like JPEG2000, TIFF, JPEG, PSD may allow metadata to be stored along with the media signal (e.g., an image). Metadata and instructions associated with a media object may be stored in the same device as the media object, or in a remote device (e.g., a remote database). In these types of applications, decoders may be programmed to extract metadata from each of the different sources. The media type of file type may signal to the decoder to extract

metadata from these different sources. Alternatively, the watermark message or file metadata may enumerate the different sources of the metadata.

In these applications, the decoder may be designed to get metadata from one or more sources and then present an aggregate of all information. The decoder may perform an aggregation function automatically or prompt the user to select desired sources of metadata for display.

Additional Functionality

A number of features can be implemented that take advantage of the watermark message payload and watermark links to other data and actions. Several examples are highlighted below:

Metadata

Metadata can be expressed in many forms and provide additional functionality. In general, metadata is information about the media object. It may also include machine instructions to be executed, or a reference to information or instructions (object, user, program, or machine identifiers, URLs, pointers, addresses, indices or keys to a database, etc.). The machine instructions may, for example, control rendering, decoding, decrypting or other processing of the object. Alternatively, the instructions may provide some ancillary functionality.

One important application of metadata is to provide ownership information. Another is to provide licensing terms and usage rights.

The metadata can be used to describe attributes of the media object that the user or other applications may use. For example, one attribute may designate the content as restricted, which prevents an application from rendering the content for unauthorized users. Another attribute may designate the object as commercial, which requires an application to seek payment or a license before the object is rendered.

Multiple Watermarks

The media object may contain two or more watermarks or watermark messages, each associated with a distinct set of information or actions. For example, the media object may contain a creator ID, a distributor ID, etc. that link to information about the creator and distributor, respectively.

There are a number of ways to add watermarks to a media object, either at object creation time, or later as the object is transferred, copied, or edited. One way is to interleave separate watermarks in different portions of the media object. This can be accomplished by modifying independent attributes of the media object. Independent, in this context, means that each watermark does not interfere with the detection of the other watermarks. While two independent watermarks may alter the same discrete sample of a media object, they do so in a manner that does not cause an invalid read of any of the watermarks.

Independent watermarks may be located in different spatial or temporal locations of the host media signal (e.g., image, video, audio, graphical model, etc.). The may also be located at different frequency bands or coefficients. Also, they may be made independent by modulating independent features of the signal, such as phase, energy, power, etc. of different portions of the signal.

To illustrate the concept, consider an example of a still image object. Each independent watermark may be defined through a different protocol, which is used to encode a different watermark message (e.g., different watermark links for a creator, distributor, user, etc. of the media object). Independent spatial watermarks may be interleaved by mapping each of the watermarks to a unique set of spatial locations in the image.

In a similar fashion independent watermarks may be encoded in a temporal data sequence, like audio or video, by mapping each watermark to unique temporal locations.

Digital Rights Management

The watermark may be linked to information and processing actions that control use of the media object. For example, the metadata may indicate the owner of the intellectual property rights in the object as well as licensing terms and conditions. Further, the watermark link or metadata trigger processing actions that control use of the object, such as requiring the user to submit payment, and exchanging decoding keys (e.g., decryption, decompression, etc.). While some amount of decoding of the object

may be required to extract the watermark, the remainder of the content may remain encoded and/or encrypted until the user obtains appropriate usage rights.

The digital rights management functionality can be implemented in a licensing or usage rights server, such as the metadata server. This server determines the owner and licensing terms based on the watermark message and executes actions required to authorize use of the object, e.g., electronically receiving payment information from the user, establishing and recording a user's assent to the license, forwarding transaction details to the owner, and returning a usage key to the user. As the user plays or renders the media object, the watermark decoder can send a message to the server to log information about the usage, such as instances of use, machine ID of the player, time of use, etc.

Watermark and Context Information for Usage Control

Watermark decoders can also use context information along with information extracted from a watermark embedded in a media object to control use of that object. For example, the watermark decoder may be programmed to control the rendering, editing, copying or transfer of a media object depending on control data in the watermark and context information derived from the device or system in which the object resides. For example, the watermark may instruct the decoder to inhibit rendering of a media object if its outside of a given file (e.g., a specified web page, computer system, computer network, etc.). After decoding the watermark including such control data, the decoder determines the pertinent context information that must be present to enable a particular operation. This may encompass such actions as verifying the presence of a user identifier, computer identifier, file identifier, storage media identifier, computer identifier, network identifier, etc. before enabling the operation. These identifiers are stored in association with the entities that they identify, or are dynamically derived from these entities. For example, a file identifier can be stored in a file header or footer, or derived from content in the file. A storage device identifier can be stored on the storage device, or derived from content on the storage device or some attribute of it.

This context sensitive control of media objects is particularly useful in controlling the use of media files like music and movies, but applies to other types of

media signals in which watermarks can be encoded. For instance, such context sensitive control can be used to prohibit rendering, copying or transfer of a media object when it is removed from the context of a web page, a computer, a storage device (e.g.., a CD or DVD), file sharing network, computers of paid subscribers of a subscription service, a collection of related media objects, etc.

Asset Management

The watermark embedded in a media object may play a role in asset management. Every time the object is processed (opened, edited, copied, etc.) an application equipped with a decoder can log information about the processing event. Alternatively, it can send a transaction event to a monitoring server via the Internet. The transaction event may specify information about the object, such as its ID, the user's ID, the time and location of use, the nature of the use (number of playbacks). The monitoring server or application records this transaction event in a database and, upon request, generates reports about the use of a given object, or by a given entity. Any of the fields of the transaction record can be used to query the database and generate custom reports.

Integration with Directory Services

Directory services like the Lightweight Directory Access Protocol (LDAP) can use the watermark or operate in conjunction with the watermark decoder to provide additional functionality. LDAP is an on-the-wire protocol used to perform directory operations such as read, search, add, and remove.

For example, an LDAP service can be used to determine when to extract the watermark link and update attributes of a media object. For example, the LDAP service may control periodic updates of the media object's attributes by invoking a watermark decoder and retrieving an update of its attributes from a metadata server at predetermined times. An LDAP search filter that includes or accesses a watermark decoder can also be provided to find watermarked media signals in files stored in file directories on computers.

Media Object Player and Delivery Integration

A watermark decoder may be integrated with software and devices for playing or editing media objects. There are a variety of commercially available players from Liquid Audio, Microsoft, and RealNetworks, to name a few. Such integration allows metadata and actions linked to the media object to be accessible to the user in a seamless fashion through the user interface of the player. For example, when a user is playing a video or audio file, a watermark enabled player may access and display linked metadata or actions automatically or at the user's selection. In either case, the user need not be aware that metadata or actions are obtained from a remote server via an object identifier. Also, such integration enables a content owner to link an object to licensing or assert usage control information or actions at playback time, whenever and wherever the object is played. Such information and actions can be implemented by placing data or instructions in a watermark within the object, or putting an object identifier in the watermark that links to metadata or actions outside the object as detailed throughout this document.

In many cases, it is desirable to access the functionality provided via the watermark transparently to the user. As such, the media player can be enhanced to display items linked via the watermark in a manner that appears to be a natural extension of the user interface. When the user or other program invokes the media player to play an object, it displays metadata or actions provided within or linked via the watermark in an extension of the media player's user interface.

These metadata or actions may be retrieved when the media player is launched or at some other time (e.g., as a background task, in response to user request, when the object is loaded to the computer or device where the media player resides, etc.). In one scenario, the media player invokes a watermark decoder on an object in response to a user's request to play the object. In this scenario, the media player passes the address of the object to the watermark decoder, which in turn, attempts to decode a watermark embedded in it. Upon locating a watermark, the decoder forwards an object identifier extracted from it to a metadata server, which returns metadata or actions associated with the object identifier. During metadata retrieval, the media player proceeds to play

the media object. When the linked metadata arrives, the extension to the media player displays returned metadata or actions.

Fig. 5 illustrates an example of an enhanced version of Microsoft's Windows Media player showing metadata and actions. In this example, the user interface window of the player is expanded in a bottom section 502 to show metadata and links associated with a media object. When the user positions the cursor over the items, "Clip," "Author," or "Copyright, the bottom tab 504 displays a URL associated with that item. By clicking on one of the items, the player invokes an Internet Browser, which sends a query to the resource at the selected URL (e.g., request an HTTP request to download an HTML document).

The watermark decoder may be invoked by another application, which launches the media player. Consider a case where a user is browsing audio or video files, either with a file browser or Internet browser. After finding a desired media object for playback, the user selects the object for playback. For example, the user could select a "Play" option via a context menu of the Windows Explorer, or via an insert in an HTML document as explained above. In the first case, a shell extension invokes the watermark decoder to get the associated metadata and executes a program (e.g., COM object, script, etc.) that runs the media player and displays the additional metadata linked via a watermark. In the second case, a Java script or other insert in the HTML document invokes the decoder and starts a program that runs the media player in a similar fashion.

One way to implement a program to control the Windows Media player is through the use of an Advanced Streaming Redirector (ASX) file. The file contains a script that launches the player and displays the metadata (e.g., URL links and information about the object) linked via the watermark. For information about ASX files and the use of these files to control Windows Media Player, see Microsoft's Developer's Network.

Content Authoring Tools

Media content authoring tools, including web page design tools, may include watermark embedding functionality to embed watermarks into content, such as web

page content. This embedded data then signals watermark decoder enabled devices and software to perform functions associated with the embedded watermark data.

Such tools may also include watermark decoding functionality to enable content developers to use the watermark decoding feature to screen media object for watermarks within the content authoring environment. If the watermark within a media object being edited conveys information (e.g., copyright owner information, licensing terms, etc.), then the authoring tool can convey this information to the user. If the watermark includes usage control information, then the authoring tool can control the use of the media object based on that information. For example, the watermark in a media object may convey an instruction that inhibits the authoring tool from editing the media object, unless the user obtains authorization from a licensing server.

Web Server Integration and Related Applications

Watermark encoding and decoding functions may also be integrated into network server application software to support functionality described in this document as well additional functionality. In particular, watermark encoding and decoding can be integrated into web servers.

Watermark encoders can be integrated into web servers to embed watermarks in media content transferred to, from or through the server. One reason for embedding watermarks at the server is to encode transaction specific information into a media object at the time of its transfer to, from or through the server. The transaction specific information relates to an electronic transaction between the server and some other computer. For example, before downloading or uploading a media object file, the server may embed information about the recipient/sender of the file into the media signal in the file (e.g., image, audio or video file). Since the watermark remains in the signal, information about the sender/recipient in the watermark remains with the media signal in the file through digital to analog –analog to digital conversion, file format changes, etc.

The server may embed a link to information or actions (links to related web sites) in the file that is uniquely tailored to the sender's/recipient's preferences. The preferences may be obtained from the user's computer, such as through popular "cookie" technology commonly used in Internet browsers, or may reside in some other

database that associates a user (a user identifier or an identifier of the user's computer) with the user's preferences (e.g., types of content preferences like news, financial information, sports, e-commerce opportunities, etc.). In this case, the server obtains the user identifier and then queries the database for the associated preferences.

The server may also use the preferences obtained in this manner to control what forms of advertising is returned or linked with the file. For example, the user may request the server to download a desired audio, video, or image file. In response, the server gets the user's preferences and downloads the requested file along with advertising information and web site links that match the user's preferences. The advertising information and links can be referenced by embedding a watermark that includes an address of the information, or that includes an index to a database entry that stores the information and/or links to other information, web sites, etc. The user's computer receiving the file downloaded from the server then renders the file and other related advertising information (e.g., provided in HTML, XML or some other conventional data format) from the server or some other server linked to the file via a watermark in the file.

The server may also embed usage control information into a watermark in a media file based on usage control rights requested by and paid for by the user in an electronic transaction between the server and the user's computer. These usage control rights can then be decoded by other applications and used to control rendering of the file, copying, recording, transfer, etc.

Network servers may also include watermark decoding functionality, such as software for decoding watermarks from media signals in files that are transferred to, from, or through the server. This enables the server to perform the many watermark-enabled functions described or incorporated into this document as well as to provide enhanced functionality. For example, the watermark may include usage control data that the server extracts and acts upon. Examples of usage control data include content rating information (adult content indicators), copy or transfer control information, rendering control information, compression/decompression control information, encryption/decryption control information, links to external information or actions, etc.

After extracting this data from the watermark, the server can modify the file based on the extracted data. For example, the server may compress or encrypt the file in a manner specified in the watermark before transferring the file. The user at the computer receiving the file would then need to have a compatible decompression or decryption program or device to render the media object in the file.

In addition to, or as an alternative to modifying the file based on the extracted watermark data, the server can send related information or instructions to the receiving computer that controls of facilitates usage of the file. For example, if the server determined from the watermark that the content was marked as "adult content", then it could send additional information with the file (e.g., HTTP header information along with a web page including watermarked content) to instruct rendering software, such as the browser, how to render the watermarked content.. The rendering software on the receiving computer can then decide how to render the content. For example, if a child is logged onto the computer receiving the file, then the rendering software can opt not to render content in the file marked as "adult content." As another example, the server may decode a watermark that instructs it to send decryption or decompression keys to the rendering software to enable the receiving computer to decrypt or decompress the content. Public key encryption schemes can be used to perform key exchanges between the sending and receiving computers. As another example, the server may decode a watermark that instructs it to send additional data along with the watermarked file including links to web sites based on information that the server decoded from the watermark.

Content Filtering and Counting

Watermark decoders can be used in computers and computer networks to filter watermarked media objects and to count instances of watermarked media objects. Filtering refers to the use of the watermark decoder to decode watermarks from objects that reside in a particular location and control their use, transfer or rendering in response to control data in the watermark, and optionally, in response to additional context data outside the watermark. These media objects may be temporarily stored at the location of the filter, as in case of a device or computer responsible for transferring the media object. Examples of such systems are e-mail servers, firewalls, routers, or

gateways in computer networks that use watermark decoding to control the transfer of certain media objects to other devices or computers based at least in part on watermarks found in the objects. The media objects may also be stored at the location of a filter on a more permanent basis. For example, the filter may be used to screen media objects that a user downloads to or uploads from a mass storage device such as a hard drive or remote personal library of music, image and movie files on a mass storage device accessible via the Internet. The filter may be used to inhibit downloading or uploading from the mass storage device in response to a watermark in a file being transferred, or alternatively, may be used to control rendering of the file.

Object counting refers to a way of logging the number of times a watermark media object is encountered, either by filtering media objects that pass through a particular device or system like a firewall or e-mail gateway, or by actively searching a network of systems like the Internet and screening for watermarked media objects found and downloaded as a result of the search. The logs maintained by watermark decoding systems can be adapted to include additional information about the object, including information from the watermark, such as an owner, user or transaction identifier, tracer data, and information about the object, such as where it was found, how it was being used, who was using it, etc. Tracer data includes data that is embedded in the file in response to some event, such as detecting unauthorized use, copying or transfer of the file.

The watermark decoder may be further augmented to send the log electronically to another device or computer in response to a specific request or in response to events. For example, the decoder can be programmed to send a report to a central database on another computer when the number of watermarked objects encountered has exceeded a threshold, and/or when certain information is found in a watermark, such as a particular identifier or tracer data that was embedded in the media object in response to detecting an unauthorized use or copying of it. Programmatic rules can be established within the decoder to specify the conditions under which watermarked media objects are filtered and counted, to specify which information is logged, and to specify when the logged information is transmitted to another computer.

The watermark based filtering and counting functions can be implemented in a variety of software applications and devices. Some examples include a network firewall, and other client, server, or peer-to-peer software applications that encounter media objects (such as operating systems, media players, e-mail readers and servers, Internet browsers, file sharing software, file manager software, etc.). One particular use of watermark based filtering, screening and counting is to monitor watermarked content sent in or as an attachment to e-mails sent between computers.

Watermark Based Spiders

Prior patent documents by the assignee of this patent application describe systems and methods of automated searching and watermark screening of media object files on computer networks like the Internet. See U.S. Patent No. 5,862,260, which is hereby incorporated by reference. The software used to perform automated searching and compiling of Internet content or links is sometimes referred to as a web crawler or spider.

As extension of the watermark based information retrieval described in US Patent 5,862,260 and marketed by Digimarc Corporation, watermark decoders can be employed in a distributed fashion to perform watermark screening and counting of watermarked media objects on networks, including the Internet. In particular, watermark decoders can be deployed at a variety of locations on a computer network such as the Internet, including in Internet search engines that screen media objects gathered by each search engine, network firewalls that screen media objects that are encountered at the firewall, in local area networks and databases where spiders do not typically reach, in content filters, etc. Each of these distributed decoders acts as a spider thread that logs watermark information as described in this document and those incorporated by reference. Examples of the types of information include identifiers decoded from watermarks in watermarked media objects, media object counts, addresses of the location of the media objects (where they were found), and other context information (e.g., how the object was being used, who was using it, etc.). The spider threads, in turn, send their logs or reports to a central spider program that compiles them and aggregates the information into fields of a searchable database.

Event Scheduling Based on Embedded Data

Watermark decoding may be used in conjunction with an event scheduler to schedule programmatic events that occur in response to decoding a watermark message of a given type. Throughout this document, there are many instances of triggering actions in response to decoding information, instructions, or links from a watermark message. In some cases, these actions are programmatic actions made by software in response to the watermark, while in other cases, these actions are device actions made by hardware circuitry, such as in the case of usage control of media signals in hardware implementations of audio and video players, recorders, etc.

Rather than taking an action immediately upon decoding a watermark message, an event scheduler can be used to schedule programmatic or device actions to occur at a later event, either at a specified time according to a clock or timer, or in response to a subsequent input, action, etc.

One example of this is to schedule a link to a web site to be activated at later, and perhaps periodic events. For example, a watermark decoder implemented in a browser, operating system, media player or other software application decodes a watermark from a media object that links the object to one or more web sites. In response to decoding the watermark, the decoder schedules programmatic actions to occur at later times. Some examples of these actions include: displaying a window with a link to a specified web site at periodic intervals or in response to a programmatic action like the launching of a browser, media player or other application. This approach can be used to prompt the user to buy a product, such as the media object (a music or video track) or some product depicted in the media object. Using this approach, many actions can be scheduled to occur in response to a single decoding of the watermark.

Integrating a Watermark Encoder in Operating Systems and Other Applications

In some applications, it will be useful to encode a watermark or overlay one or more additional watermarks to perform any of the functions mentioned above (e.g., to track uses, refresh usage rights, add links to additional information and actions, etc.).

Watermark encoding functionality could be added to an operating system, Internet browser, or other applications. For instance, through a drag and drop procedure, a user could embed a watermark in a media object as a means of enabling the various functions outlined above.

As another example, a watermark encoder may be integrated in a file browser, Internet browser, media player, or other application using the same integration techniques outlined above for the decoder. Fig. 1, for example, shows watermark encoder functionality integrated into the Windows Explorer file browser via a shell extension. In particular, the watermark encoder is implemented as a shell extension handler. This handler is registered as a context menu extension in the registry of the Windows Operating system. As an alternative, it could be implemented as a properties page extension handler.

To access the watermark encoder in the Fig. 1 example, the user right clicks on a media object, and selects the context menu option called "Embed Information." In response, the handler displays the window 600 shown in Fig. 6. This window enables the user to enter various Ids (e.g., a creator ID, image ID), which are encoded into an image via a watermark. The user may also set or select attributes of the image object. Finally, the user can control the embedding process by adjusting the durability of the watermark through a scroll bar control. The user can compare the original and watermarked versions of the object by selecting the "Original" and "Watermarked" tabs. When satisfied, the user can save the watermarked image and exit the window (e.g., by selecting close). Metadata and actions may be associated with the image object by forwarding them to the metadata server, which associates them with an object ID.

While the example in Fig. 6 depicts a still image object, a similar approach may be used to embed watermarks in other media objects, such as video and audio objects. To compare marked and unmarked audio or video objects, the shell extension may be designed to launch a media player to play the marked and unmarked objects as desired.

Digital Watermarks as a Gateway and Control Mechanism

The Internet presents security challenges to corporations and others who have confidential information and connections to the Internet or other types of computer

networks. Traditionally, documents containing confidential information are marked with a legend or other visual indicia with words such as "CONFIDENTIAL", "PROPRIETARY", etc. The presence of these marks alert anyone handling such documents that they should only be transferred outside of a company's control under special precautions. It is relatively difficult and unusual for someone to manually send such a document inadvertently to an unauthorized receiver. However, the use of electronic communication services, such as e-mail and file transfers over the Internet, makes malicious and inadvertent transfer trivial.

The Internet, electronic mail, and other network file transfer protocols (FTP, HTTP, etc.) simplify and speed the communications process. Such protocols also make it much easier for someone to inadvertently or accidentally send a confidential document to an unauthorized receiver.

This section describes a system and related methods for using digital watermarks to control the transmission and/or receipt of documents transmitted over computer networks such as the Internet. The system and methods can be used to prevent the accidental dissemination of information to unauthorized receivers. Furthermore, while no security system is fool-proof, the system and methods help guard against the intentional, but unauthorized, dissemination of information to unauthorized receivers.

Most electronically transmitted messages contain text. However, electronic mail systems generally allow images (i.e. pictures) or sound bites to be embedded into and form part of a message. For example, a message can contain a "stamp" with the word "confidential" or a message can contain a sound clip with the word "confidential". An image or sound clip that forms part of an electronic message can carry a digital watermark that can be detected and read by digital watermark reading programs. In addition, text may carry a steganographic information, such as by adding or removing characters or lines according to a hidden pattern that does not change the meaning of the text, such as a pseudorandom pattern created by spread spectrum modulating a binary message with a PN sequence. Such a pattern may be used to add or remove spaces at the ends of lines and paragraphs in a desired pattern. Alternatively,

an image of a text document can carry a watermark by toggling halftone dots on or off according to watermark pattern.

The "payload" or digital data in a digital watermark or steganographic message typically has a number of different fields. One or more of these fields can be dedicated to a flag which indicates that the document or image containing the message is confidential or otherwise classified and that it should only be disseminated in a particular manner.

Typically, e-mail enters a transmission network by way of an e-mail server. Programs that can detect and read watermarks are well known and commercially available. In one implementation of our system, the e-mail server passes each e-mail message through a watermark detection and reading program prior to sending the message out over a network. If the watermark program detects a watermark, it interrogates certain flag bits to determine how the message should be handled. If the watermark reading program finds that a particular flag is set, it can take action such as alerting both the sender and a network administrator. If the watermark program finds no watermark or finds that a particular flag is not set, the message is sent over the network in a conventional manner.

The system can serve as a control mechanism for controlling the dissemination and receipt of electronic messages.

Messages and documents also enter the Internet and other electronic networks received from servers such as Web servers and FTP servers. In a similar fashion a watermark detection program can interrogate documents on servers such as Web and FTP servers and take action such as sounding an alarm if a watermark with a particular flag in a particular state is detected. Such a system may be used to screen files on a web server that have been unintentionally or inadvertently added to a web site. Unauthorized files detected by a programmatic screen of the file directories may be deleted or prevented from being transferred in response to requests for the files via FTP or HTTP requests.

A first embodiment of the system is designed to monitor e-mail messages transmitted over the Internet. This first embodiment is used to prevent the accidental dissemination of confidential e-mail messages to unauthorized users. The first

embodiment prevents the transmission of confidential documents to anyone. Alternate embodiments control the transmission of confidential documents to unauthorized users or for unauthorized purposes. For example, it is easy to add addressees to an e-mail message. Someone may address an e-mail message that includes confidential information to a large group of people without realizing that one of the addressee is not authorized to receive confidential information. The system will prevent such an e-mail from being transmitted to the unauthorized person even though the sender included the address of this person in the list of addressees.

A typical confidential document 10 is represented in Figure 8. The document 10 can either be an e-mail message, or alternatively it may be a document that is attached to an e-mail message. The document 10 includes a confidentiality stamp 11 and lines of text. The confidentiality stamp 11 is an image that has the word "confidential" superimposed over a background that has a variety of lines. That is, the background in image 11 has lines of a the width of which are varied to carry a watermark in accordance with the teachings of US application 09/074,034, filed May 6, 1998 (which corresponds to PCT application PCT/US99/08252), and US application 09/127,503, filed July 31, 1998 (which corresponding to PCT application PCT/US99/14532). The disclosures of the above referenced patent applications are hereby incorporated by reference. Alternatively the background of image 11 or other background portions of the document 10 may comprise a weave or tint pattern that carries a digital watermark signal, such as pseudorandom image pattern formed by spread spectrum modulating a binary message with a PN sequence and converting the modulated sequence to an image of multilevel per pixel values. In still another alternative embodiment instead of having an image 11 embedded in the massage, the message may contain an audio clip with the word confidential. The audio clip would be watermarked using audio watermarking techniques. However, in the first embodiment described herein the, image 11 has both a human readable word "Confidential" and a digital watermark that can be read by a watermark detection and reading program.

The data fields and flags in a typical watermark payload are shown in Figure 9. It should be understood that the fields and flags shown are merely representative and

they can take many alternative forms. The system utilizes one of the flag fields to indicate that a particular document is confidential. The other fields can be used in a conventional manner.

Fig. 10 shows an e-mail system with watermark detecting and reading routines for monitoring and filtering content. A relatively large number of individual user terminal 1301a to 1301e are connected to an e-mail server 1302. Such connections can be by an Ethernet LAN, by dial up modems, or other conventional wire or wireless computer network connection. The e-mail server 1302 has an interface 1303 to the Internet 1304 and it receives and sends messages from the individual users' terminals to the Internet. The e-mail server 1302 itself is conventional, but has a programmatic interface for enabling watermark detecting and reading of messages processed by the server. With the present invention, before the e-mail server 1302 transmits a message from one of the individual User Terminals 1301a to 1301e to the internet, the e-mail server passes the message through a watermark detection and reading program 1305. Both the e-mail message itself and any attached documents are passed through this program. The watermark detection and reading program 1305 determines if a message includes a digital watermark. If a watermark is detected, the confidentiality flag bit is interrogated. If the flag bit is set to "confidential", this embodiment returns the message to the sender. Thus, the first embodiment prohibits any confidential information from being transmitted as part of an e-mail message.

A second embodiment provides for a wider array of alternatives. As shown in Fig. 11, the second embodiment includes a database 1401. The database 1401 includes a list of different groups of potential message senders, a list showing different groups of potential message recipients and a set of possible categories indicated by the setting of the various flags in a message. For example, the senders may fall into three designated sender groups S1, S2 and S3. The potential recipients can fall into three groups designated R1, R2, and R3. The database 1401 and the associated logic 1402 can implement logic rules such as indicated by the following table:

Sender Group	Recipient Group	Flag Conditions	Action
S1	R1	011	Send message
S1	R2	110	Do not sent message notify the administrator
S1	R2	001	Send message, and log fact that S1 sent a message to R2.
S1	R2	101	Return message to sender
S2	R1	011	Send message
S2	R3	110	Do not sent message and notify the system administrator

The system includes a buffer 1403 that buffers messages passing through the server in a FIRST-IN, FIRST OUT fashion. This buffer may store the messages themselves or pointers to their locations within the server's file directory structure. A watermark reading program 1404 reads the message and/or attachments, attempts to detect a digital watermark or embedded steganographic message, and if detected, provides pertinent message payload information to a database interrogation program 1405. A sender/recipient filter program 1402 implements control logic that filters the sender and recipient addresses and maps them to previously assigned groupings. This program provides information for a particular message to the interrogation program 1405, which in response, uses the message payload flags and sender/recipient codes to look up the associated control action associated with the message. The system sends the control action to a message distribution control mechanism 1406 that programmatically instructs the server to take the particular action with a message in the buffer 1403.

In summary, the system considers the message sender, the message recipient and the condition of the flags to determine what action should be taken. A system can include many alternatives in addition to those shown above, or a system might include only a few alternatives. For example, the system could include only a list of addresses which are authorized to receive messages which have a confidentiality flag set to "confidential". Such a system would allow confidential documents to be sent only to selected addresses. Alternatively or in addition, the system could include a list of

individuals authorized to send confidential documents. The system could merely check the sender against this list or alternatively, the system could require that a password be entered when such messages are encountered.

It should be noted that the above table gives only representative rules. In an actual system there could be many more groups and some groups might contain a single individual. Furthermore the table could have many more lines. The table above shows only three flag bits. A system could have more or less flag bits as the needs of the particular system require.

In alternate embodiments, the confidentiality stamp could include a watermark in an image by means other than using line width or halftone dot modulation as described above. The background of the stamp could include an image carrying a digital watermark.

In an alternative embodiment, rather than checking for a digital watermark, the system could check for a text string such as "confidential" and take action in response to locating such a text string. Also, the system could check for a hidden steganographic pattern of spaces at the end of the lines or paragraphs.

While the previously described embodiments apply to e-mail systems, similar precautions could be taken with FTP servers or with Web servers.

Content Filtering and Indexing System Using Digital Watermarks

Fig. 12 is a diagram illustrating a content filtering and indexing system. In this system, a digital watermark embedder embeds digital watermarks into content such as images, video or audio, so that the digital watermark is imperceptible or substantially imperceptible in the content when rendered to end-users (e.g., printed, displayed, played, etc.). This form of embedding involves perceptually adapting the embedded watermark signal to the host media signal based on human perceptual models. The embedder, referred to as a watermark content marker 1502, 1504 is incorporated into content servers, such as web servers, streaming media servers, FTP servers, etc. (1506, 1508) and marks content on those servers. The embedder preferably embeds watermark payloads repeatedly throughout the content, such that different temporal or spatial portions carry the same or different payload relating to the content in which it is

embedded. For example, content type flags may change as the content type changes from advertising to programming, and from one rating to another (G, PG, R, X according to content type).

This content is then sent via a communications network 1510 to other systems, such as web browsers 1512, email readers 1514, media players, etc. Along the communication path, the content may pass through a firewall 1516 and LAN 1518. Digital watermark reader programs, shown as watermark content filters 1520, 1522, and 1524, scan the content for digital watermarks, and if found, decode the message payload of these watermarks. The message payload includes information about the content owner (owner ID), distributor (distributor ID), and/or transaction (transaction ID, receiving machine ID, IP address, sender/recipient address, or user ID). In addition, the payload includes content flags that indicate content type, such as adult content ratings, limited access ratings, etc.

The watermark content filters filter content by controlling its transfer or rendering in response to the content flags. In addition, they send the embedded IDs and content type information to a searchable database 1530 that logs information about each filtering event in the database. The system logs additional event information that is derived from out-of-band data like a file header or the machine where the content is detected, such as network address of detection event, sender and recipient address, user name, etc. The content filters implement a set of rules to control transfer and rendering of content. For example, the content filter for firewalls prevents transfer of content files with certain flags or combinations of flags set in the watermark payload. The firewall may be used to prevent transfer of adult content onto the company network, for example.

The content filters for web browsers, email readers and media players prevent rendering of the content based on the content flags and rules established in the software application where the filter is integrated. For example, user defined rules may be used to block rendering of adult content on computers used by children. These rules are activated when a user logs onto the application, supplying a user ID indicating to the filter that a particular user is using the application and certain rules applicable to that user ID are activated.

A similar approach can be used to filter advertising from content for users that have paid higher subscription fees for content. In this case, content including advertising flags is detected via the watermark and removed before rendering. Similarly, certain types of content can be blocked from being rendered on user's machines where the user's subscription does not authorize playback of content with a particular type of content flags embedded in it. The content filter can be used to enable or block rendering of content based on the user's subscription rights and the flags in the content. This approach is particularly suited for distributed peer to peer models where users may receive content from many different peer computers.

The content filtering system also enables the searchable database to keep a searchable index of content indicating computer addresses (URLs or IP addresses) where copies of the content can be found. As such, the system can be used to create a search engine accessible via the Internet to enable users to search the database by content type, content ID, distributor, or other attributes linked to the content ID and find content items and their corresponding addresses.

Fig. 13 is a diagram illustrating a distributed watermark detector system. In this system, digital watermark detectors (1602, 1604, 1606, 1608) are implemented as spider thread programs that screen content for watermarks, and when found, report information from the watermark payload and information about the event (network address of content, user information, time and date, etc.) to a central spider program 1610 executing on server attached to the Internet 1612. The thread programs are implemented in internal systems 1602, such as content databases on company's LANs, firewalls 1604 on LANs, content filters 1606 within LANs or software applications like browsers or operating systems, and search engines 1608, such as web crawlers that fetch vast amounts of linked web pages and index the pages based on their contents.

The spider program 1610 stores the watermark payload and event information in a database 1612. The spider program also cross references the database information compiled from watermark detection events to additional databases that associate content IDs from the watermark payloads with content attribute information, such as the content title name, owner name, artist, lyrics, URLs of related web sites, etc. This enables users to access the spider program, search for content based on title or some

other attribute, and then cross reference to network addresses where the content can be found.

This system enables deep linking of content on devices across vast and complex networks. The system may be adapted for devices like computers connected to the Internet, music appliances linked on a music file sharing network, set-top boxes and personal video recorders linked on a video file sharing network, etc. In some systems, the nodes of the network where the distributed spiders are located also act as content servers. When a user finds a piece of content via an automated search of the searchable database 1612, the user can send a request to the content server where the content is located and request transfer of the content file from the content server to the user's computer or other device, such as music appliance, set top box, wireless phone, etc.

There are a number of potential enhancements or modifications to the distributed content filtering and spidering systems described above. When filtering content based on the watermark, the watermark payload can include content type flags or an index to content type descriptors stored in a database. In the latter case, the system queries the descriptor database to determine the descriptors that define the nature of the content. Either the content flags or the descriptors can be used by search engines to index content distributed on the network by content type.

To expand the spider's database in the system of Fig. 13, the system may also include programs that perform a automated automated crawl of linked web sites, screen the sites for watermarked content, and then use the content flags in the payload or the content descriptors indexed by the payload to create a searchable index of the content by its type. Users can then search this index to find content matching content type criteria, and get a list of content matching that criteria along with the content's address.

The manner in which the filters control rendering may vary. For example, the watermark filter may prevent playback entirely, scramble the playback, or only scramble an objectionable portion based on the content type flags or descriptors. Since watermarks are repeated throughout frames of the content, the filter can offer fine grained control over which frames of audio or video get scrambled or blocked, or which portions of a still image get blocked. In addition, the filter can selectively control the type of content that gets rendered, for example, by swapping in frames of

content based on user demographic, such as age, location, etc., based on user preferences supplied by the user. In order to swap content, the content delivery mechanism can stream alternative frames in response to the filter's control signals.

The watermark filter may use the watermark payload to link to related information. Initially, the watermark reader links content to a default network resource, such as a default web page. In particular, the watermark payload includes an identifier that indexes a database entry. This entry includes the URL of a default web page. However, the content owner/distributor can dynamically update the database entry to specify a new machine behavior for the identifier, such as a new URL specific to the owner's web site.

Operating Environment for Computer Implementations

Figure 7 illustrates an example of a computer system that serves as an operating environment for software implementations of the watermarking systems described above. The embedder and detector implementations are implemented in C/C++ and are portable to many different computer systems. Fig. 7 generally depicts one such system.

The computer system shown in Fig. 7 includes a computer 1220, including a processing unit 1221, a system memory 1222, and a system bus 1223 that interconnects various system components including the system memory to the processing unit 1221.

The system bus may comprise any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using a bus architecture such as PCI, VESA, Microchannel (MCA), ISA and EISA, to name a few.

The system memory includes read only memory (ROM) 1224 and random access memory (RAM) 1225. A basic input/output system 1226 (BIOS), containing the basic routines that help to transfer information between elements within the computer 1220, such as during start-up, is stored in ROM 1224.

The computer 1220 further includes a hard disk drive 1227, a magnetic disk drive 1228, e.g., to read from or write to a removable disk 1229, and an optical disk drive 1230, e.g., for reading a CD-ROM or DVD disk 1231 or to read from or write to other optical media. The hard disk drive 1227, magnetic disk drive 1228, and optical disk drive 1230 are connected to the system bus 1223 by a hard disk drive interface

1232, a magnetic disk drive interface 1233, and an optical drive interface 1234, respectively. The drives and their associated computer-readable media provide nonvolatile storage of data, data structures, computer-executable instructions (program code such as dynamic link libraries, and executable files), etc. for the computer 1220.

Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and an optical disk, it can also include other types of media that are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, and the like.

A number of program modules may be stored in the drives and RAM 1225, including an operating system 1235, one or more application programs 1236, other program modules 1237, and program data 1238.

A user may enter commands and information into the personal computer 1220 through a keyboard 1240 and pointing device, such as a mouse 1242. Other input devices may include a microphone, sound card, radio or television tuner, joystick, game pad, satellite dish, digital camera, scanner, or the like. A digital camera or scanner 43 may be used to capture the target image for the detection process described above. The camera and scanner are each connected to the computer via a standard interface 44. Currently, there are digital cameras designed to interface with a Universal Serial Bus (USB), Peripheral Component Interconnect (PCI), and parallel port interface. Two emerging standard peripheral interfaces for cameras include USB2 and 1394 (also known as firewire and iLink).

In addition to a camera or scanner, watermarked images or video may be provided from other sources, such as a packaged media devices (e.g., CD, DVD, flash memory, etc), streaming media from a network connection, television tuner, etc. Similarly, watermarked audio may be provided from packaged devices, streaming media, radio tuner, sound cards, etc.

These and other input devices are often connected to the processing unit 1221 through a port interface 1246 that is coupled to the system bus, either directly or indirectly. Examples of such interfaces include a serial port, parallel port, game port or universal serial bus (USB).

A monitor 1247 or other type of display device is also connected to the system bus 1223 via an interface, such as a video adapter 1248. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The computer 1220 operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 1249. The remote computer 1249 may be a server, a router, a peer device or other common network node, and typically includes many or all of the elements described relative to the computer 1220, although only a memory storage device 1250 has been illustrated in Figure 7. The logical connections depicted in Figure 7 include a local area network (LAN) 1251 and a wide area network (WAN) 1252. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the computer 1220 is connected to the local network 1251 through a network interface or adapter 1253. When used in a WAN networking environment, the personal computer 1220 typically includes a modem 1254 or other means for establishing communications over the wide area network 1252, such as the Internet. The modem 1254, which may be internal or external, is connected to the system bus 1223 via the serial port interface 1246.

In a networked environment, program modules depicted relative to the personal computer 1220, or portions of them, may be stored in the remote memory storage device. The processes detailed above can be implemented in a distributed fashion, and as parallel processes. It will be appreciated that the network connections shown are exemplary and that other means of establishing a communications link between the computers may be used.

Extensions to Other Forms of Media Object Linking

The approaches described above can be implemented for a variety of media object files, including image, graphics, video and audio files, or files containing two more different media types. Also, media objects may be linked to their metadata via data structures other than a watermark embedded in the object. For example, the object identifier need not be inserted in a watermark, but instead may be placed somewhere else in the media object file, such as a file header. Such an identifier may be inserted

into the header of coded or compressed files. To extract the identifier, a decoder parses the header and extracts the object identifier. Then, the decoder forwards the identifier to a metadata server, either directly, or by launching another application, such as web browser, to issue the metadata request and output the data and/or interpret code returned from the metadata server.

Concluding Remarks

Having described and illustrated the principles of the technology with reference to specific implementations, it will be recognized that the technology can be implemented in many other, different, forms. To provide a comprehensive disclosure without unduly lengthening the specification, applicants incorporate by reference the patents and patent applications referenced above.

The particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.

In view of the wide variety of embodiments to which the principles of the invention can be applied, it should be recognized that the detailed embodiments are illustrative only and should not be taken as limiting the scope of the invention. Rather, we claim as our invention all such embodiments as may come within the scope and spirit of the following claims, and equivalents thereto.

We Claim:

1. A file browser system comprising:

a file browser for displaying in a user interface a representation of media object files stored in memory; and

a file browser extension for decoding an object identifier from a selected media object file and for displaying in an extension of the user interface metadata or an action associated with the media object file via the object identifier.

2. The file browser system of claim 1 wherein the object identifier is decoded from a watermark embedded in the selected media object file.

3. The file browser system of claim 1 wherein the file browser extension displays the metadata or action in a context menu extension of the user interface of the file browser.

4. The file browser system of claim 1 wherein the file browser displays the metadata or action in a property page extension of the user interface of the file browser.

5. The file browser system of claim 1 wherein the file browser extension forwards the object identifier to a metadata server, and displays metadata or an action returned from the server.

6. The file browser system of claim 5 wherein the file browser extension extracts and displays metadata from the media object file along with metadata returned from the metadata server.

7. The file browser of claim 1 wherein the metadata or action is displayed as a URL link to information or a program associated with the selected media object file.

8. A file browser system comprising:

a file browser for displaying in a user interface media object files stored in memory; and

a file browser extension for encoding an object identifier into a selected media object file and for displaying in an extension of the user interface one or more options for enabling a user to enter input to control the encoding of the object identifier.

9. The method claim 8 wherein the file browser extension comprises a watermark encoder for encoding the object identifier into the selected media object file.

10. A watermark decoder system comprising:

a host application having a user interface for displaying a representation of media object files; and

an extension to the host application for decoding a watermark from a selected media object file and for displaying in an extension of the user interface metadata or an action associated with the media object file via the watermark.

11. An internet browser on a computer readable medium, the browser comprising:

a listener program for identifying a media object in an HTML document; and for inserting a handler into the HTML document when an object identifier is extracted from the media object;

wherein the handler is operable to display metadata linked via the object identifier in response to user input.

12. The internet browser of claim 11 wherein the object identifier is decoded from a watermark embedded in the media object.

13. The internet browser of claim 11 wherein the metadata is retrieved from a metadata server by sending the object identifier to the metadata server.

14. A method of rendering a media object comprising:

decoding an object identifier from the media object;

sending the object identifier to a metadata server;

receiving a brand identifier from the metadata server; and

displaying a representation of the brand identifier.

15. The method of claim 14 wherein the object identifier is decoded from a watermark embedded in the media object.

16. The method of claim 14 wherein the media object is a video or an image, and the representation of the brand identifier is a graphic superimposed on a rendering of the video or image.

17. The method of claim 16 wherein the graphic is a hot link to information or an action associated with the media object.

18. The method of claim 17 wherein selecting the hot link causes retrieval of the information or action from a remote server.

19. A method for extending a user interface of a media player comprising:

in response to input requesting playback of a media object, extracting an object identifier from the media object;

using the object identifier to look up metadata associated with the media object;

extending a user interface of a media player to include a representation of the metadata associated with the media object.

20. The method of claim 19 wherein extracting the object identifier includes decoding the object identifier from a watermark embedded in the media object.

21. An electronic messaging system including a mail server which sends and receives messages, said mail server including a reading program which reads

steganographic data hidden in said messages and which controls distribution of said messages in response to data in said data.

22. A system which includes an e-mail server connected to the Internet means for transmitting messages from individual user to said e-mail server, watermark detecting means for detecting and reading watermarks in e-mail messages before such messages are transmitted from said e-mail server to the Internet, means for preventing the transmission of messages from said e-mail server to the Internet if said watermark detecting means detects a watermark which has an indication that the message including said watermark is confidential.

23. A system for controlling distribution of electronic messages that include confidential information, each electronic message having confidential information including a digital watermark carrying data that indicates that the message is confidential, a server which transmits and receives messages, said server including a watermark reading program which reads watermarks in messages and controls the distribution of such messages in accordance with the data carried by any watermarks in the messages.

24. A method of controlling the distribution of electronic messages that include confidential information, said messages including digital watermarks which carry data indicating that the message includes confidential information,

reading watermarks in messages prior to transmission of said messages, and

controlling the distribution of each electronic message which includes a watermark in response to the data carried by the watermark in the message.

25. A method of transmitting electronic messages from a sender to a receiver which comprises,

detecting and reading digital watermarks embedded in such messages to determine how the flags in such watermarks are set,

interrogating a database to determine what action should be taken with a message based upon the identity of the sender, the identity of the receiver and the flag settings in the watermark in the message.

26. A content filtering system in which watermark content filters are distributed in programs executing on computers interconnected via a communications network, the watermark content filters being operable to detect digital watermarks in content entering a location in the network, to read watermark payload information, and to take action to control rendering or transfer of the content based on content type flags in the watermark payload information.

27. The system of claim 26 wherein the content filters report detection events to a searchable database enabling users to search for content and find a network address indicating where the content has been detected by the content filters.

28. The system of claim 26 wherein the content filters execute rules that control rendering or transfer of content based on a user ID and content type flags in the digital watermarks.

29. The system of claim 26 wherein content type flags are used in combination with rules relating to a user's access rights to enable or disable rendering of advertising along with rendering of the content.

30. A distributed watermark spider system comprising:

digital watermark detectors implemented in spider thread programs executing in devices distributed on a communications network; the spider thread programs operable to screen content at corresponding devices for watermarks, and when found, report information from watermark payloads encoded into the watermark and information about the event to a spider program executing on device connected to the network.

31. The system of claim 30 wherein the information about the event includes a network address of detected content enabling remote devices to access the detected content.

32. The system of claim 31 wherein the spider program is in communication with one or more searchable databases storing the watermark payload information, information about the events, and information about the content cross referenced to the content, wherein the searchable database enables users to search for content by name and find related information about the content, including an address where the information can be fetched via the system.

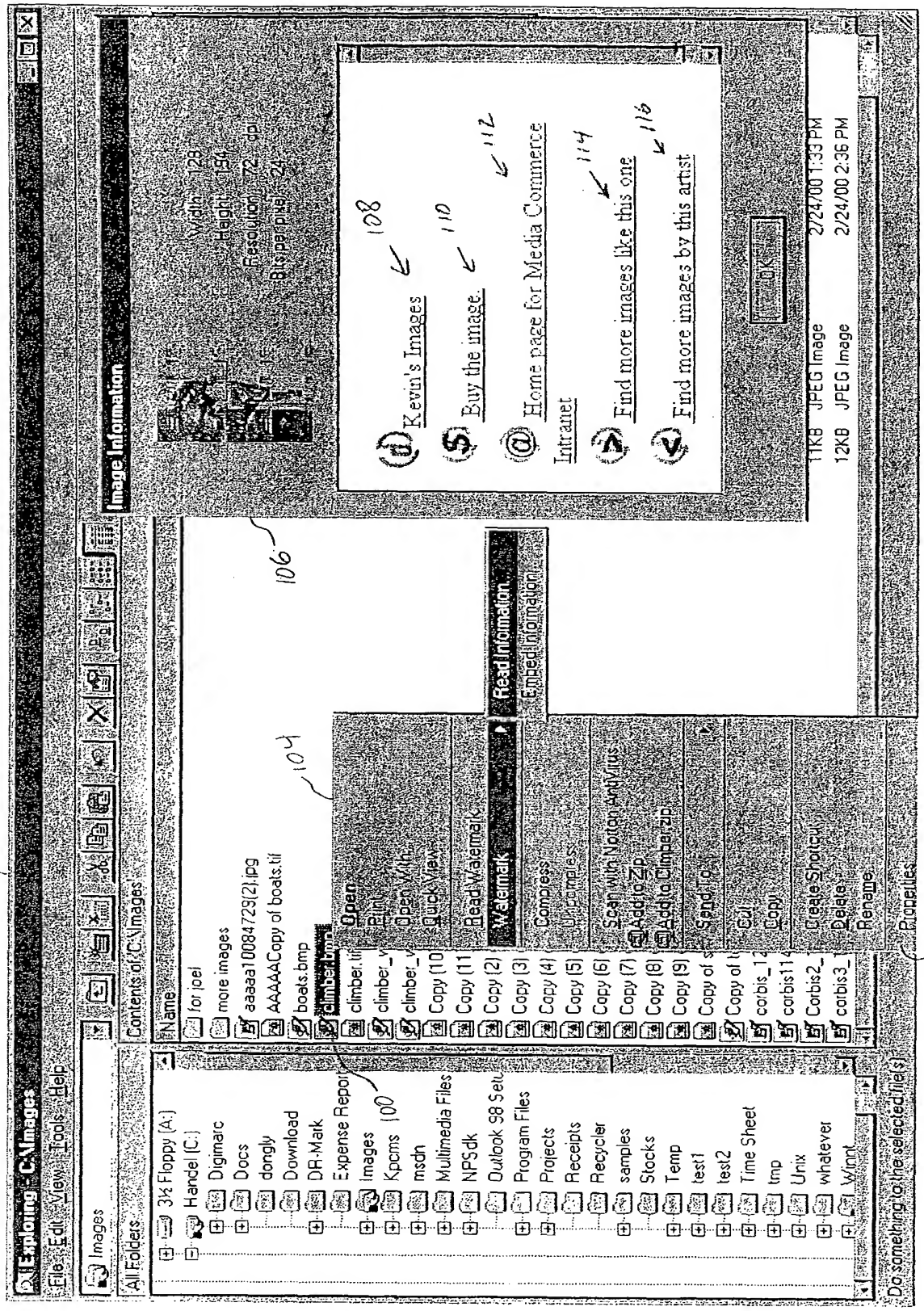

Exploring - C:\Images
File Edit View Tools Help
Images
All Folders
Contents of 'C:\Images'
Name
Image Information
Width: 128
Height: 154
Resolution: 72 dpi
Bits per pixel: 24
Kevin's Images
Buy the image.
Home page for Media Commerce
Intranet
Find more images like this one
Find more images by this artist
OK
11KB JPEG Image 2/24/00 1:33 PM
12KB JPEG Image 2/24/00 2:36 PM
Open
Read Watermark...
Watermark...
Read Information...
Embed Information...
Compress
Uncompress
Scan with Norton AntiVirus
Add to Zip
Add to Climber.zip
Send To
Cut
Copy
Create Shortcut
Delete
Rename
Properties
Do something to the selected file(s)
100
102
104
106
108
110
112
114
116
118

Fig. 1

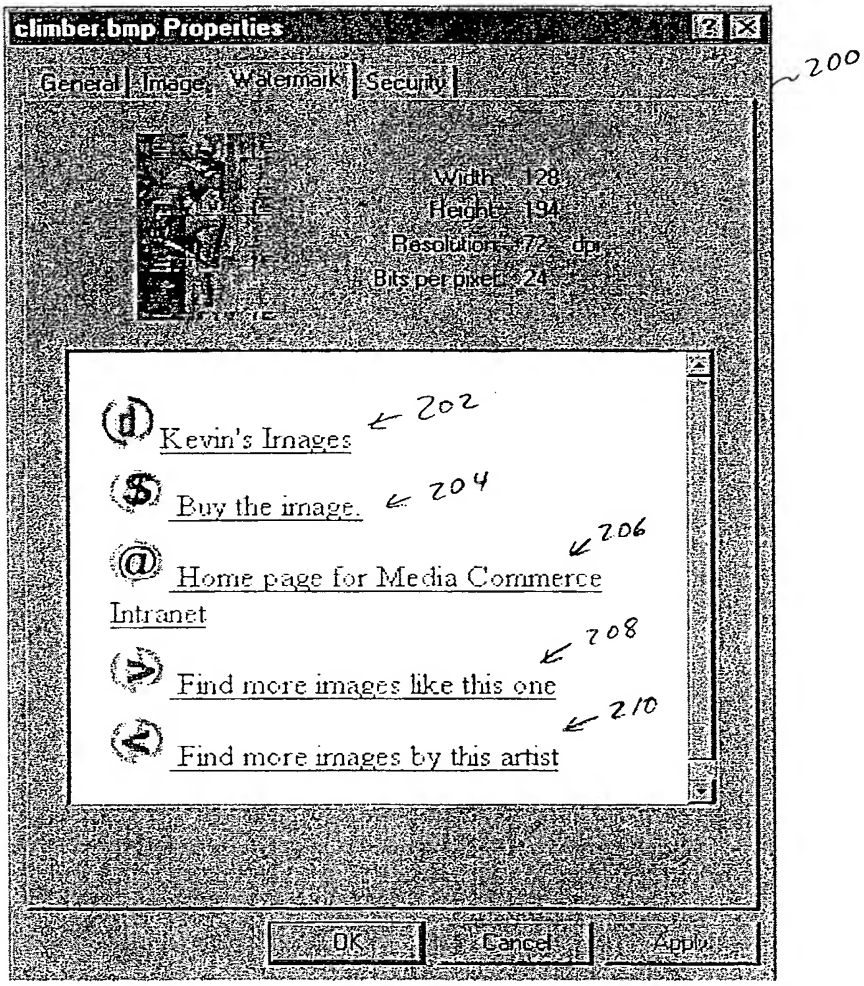

climber.bmp Properties
General
Image
Watermark
Security
Width: 128
Height: 194
Resolution: 72 dpi
Bits per pixel: 24
Kevin's Images
Buy the image.
Home page for Media Commerce Intranet
Find more images like this one
Find more images by this artist
OK
Cancel
Apply
200
202
204
206
208
210

Fig. 2

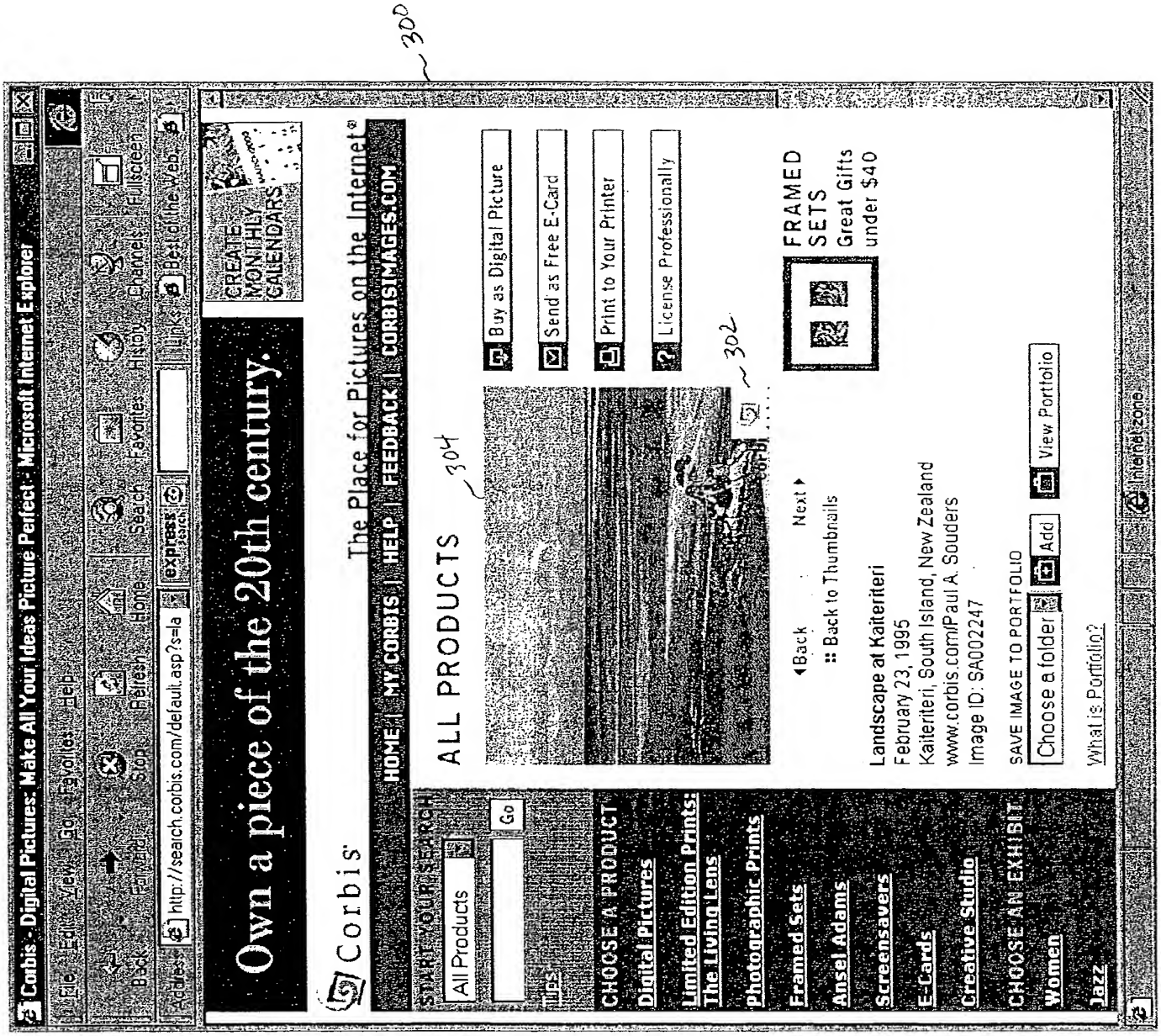

300
Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
Own a piece of the 20th century.
Corbis
The Place for Pictures on the Internet
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
CREATE MONTHLY CALENDARS
START YOUR SEARCH
All Products
Go
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints:
The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
ALL PRODUCTS
304
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
302
Back
Next
Back to Thumbnails
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
FRAMED SETS
Great Gifts under $40

Fig. 3

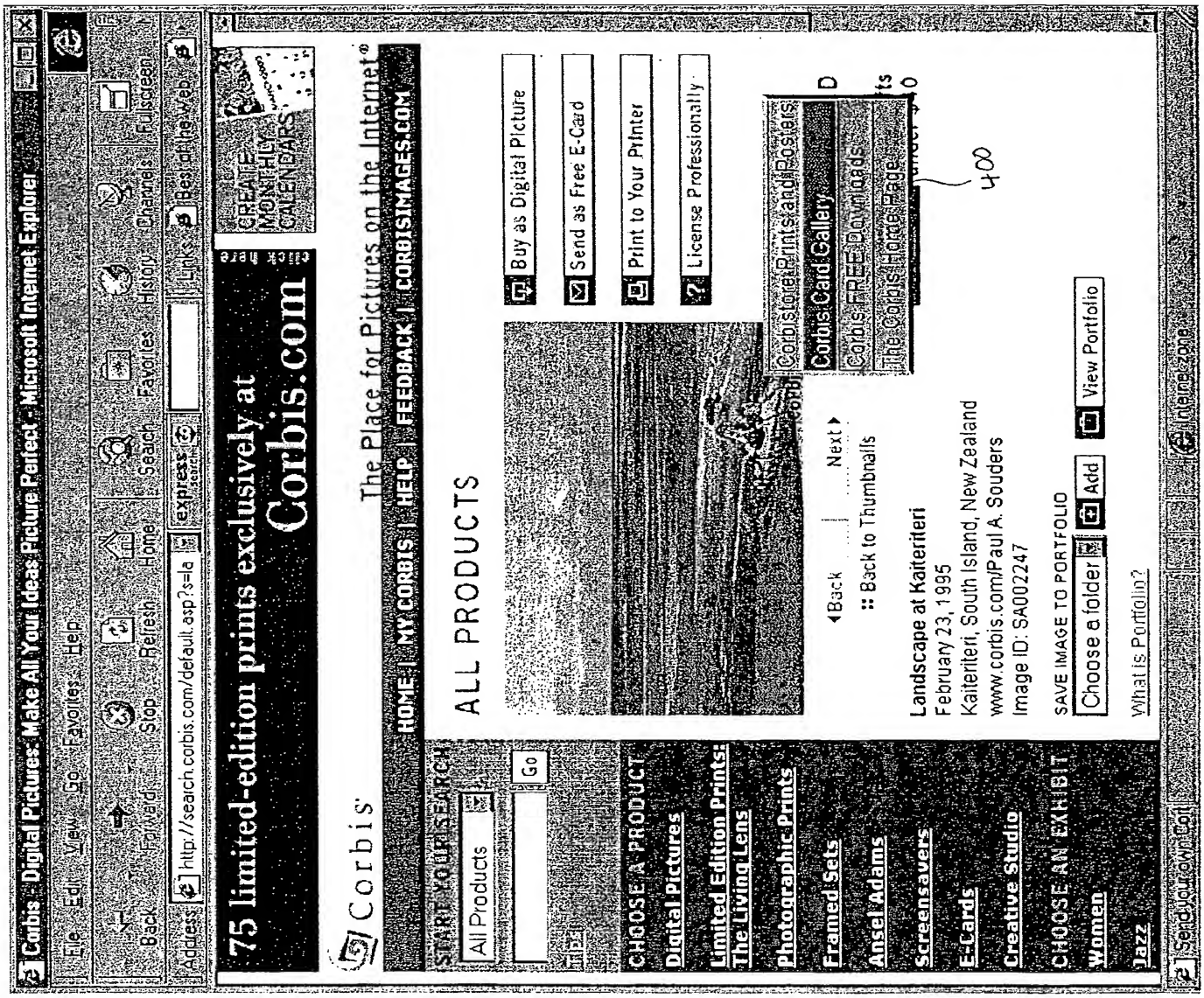

Corbis - Digital Pictures: Make All Your Ideas Picture Perfect - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen
Address http://search.corbis.com/default.asp?s=la
express search
Links Best of the Web
75 limited-edition prints exclusively at Corbis.com
click here
CREATE A MONTHLY CALENDARS
Corbis
The Place for Pictures on the Internet
HOME | MY CORBIS | HELP | FEEDBACK | CORBISIMAGES.COM
START YOUR SEARCH
All Products
Go
CHOOSE A PRODUCT
Digital Pictures
Limited Edition Prints
The Living Lens
Photographic Prints
Framed Sets
Ansel Adams
Screensavers
E-Cards
Creative Studio
CHOOSE AN EXHIBIT
Women
Jazz
ALL PRODUCTS
Buy as Digital Picture
Send as Free E-Card
Print to Your Printer
License Professionally
Back Next
:: Back to Thumbnails
Corbis Prints and Posters
Corbis Card Gallery
Corbis FREE Downloads
The Corbis Order Page
400
Landscape at Kaiteriteri
February 23, 1995
Kaiteriteri, South Island, New Zealand
www.corbis.com/Paul A. Souders
Image ID: SA002247
SAVE IMAGE TO PORTFOLIO
Choose a folder
Add
View Portfolio
What is Portfolio?
Send your own Corbis
Internet zone

Fig. 4

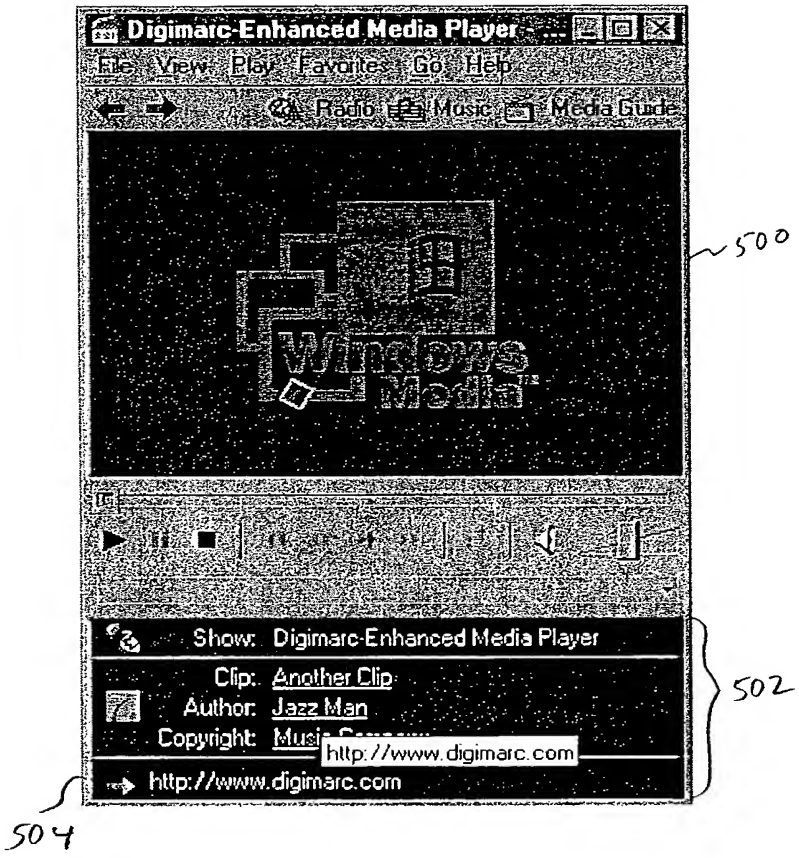

Digimarc-Enhanced Media Player
File View Play Favorites Go Help
Radio Music Media Guide
Windows Media
Show: Digimarc-Enhanced Media Player
Clip: Another Clip
Author: Jazz Man
Copyright:
http://www.digimarc.com
http://www.digimarc.com
500
502
504

Fig. 5

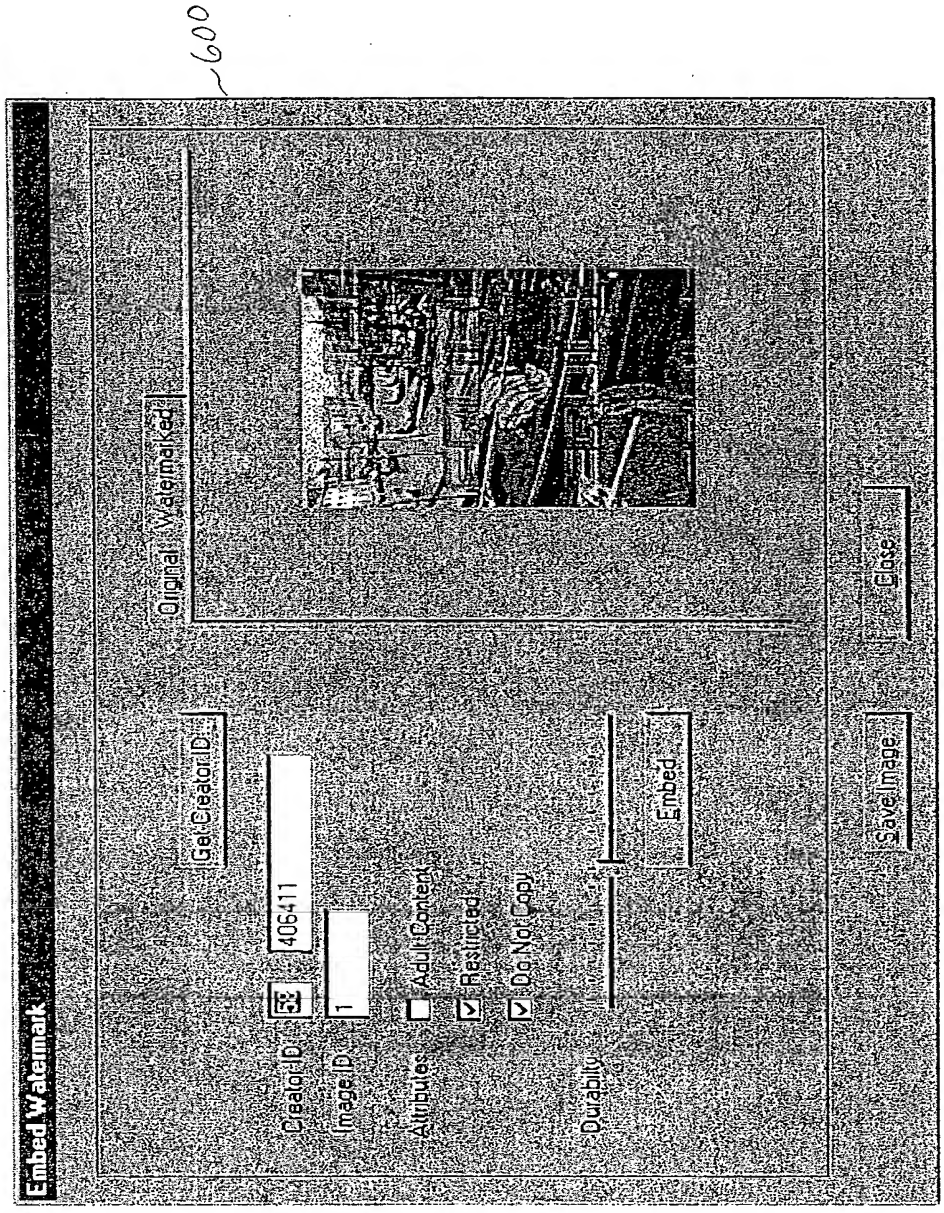

600
Embed Watermark
Original
Watermarked
Get Creator ID
406411
Creator ID
Image ID
1
Attributes
Adult Content
Restricted
Do Not Copy
Durability
Embed
Save Image
Close

Fig. 6

FIG. 7

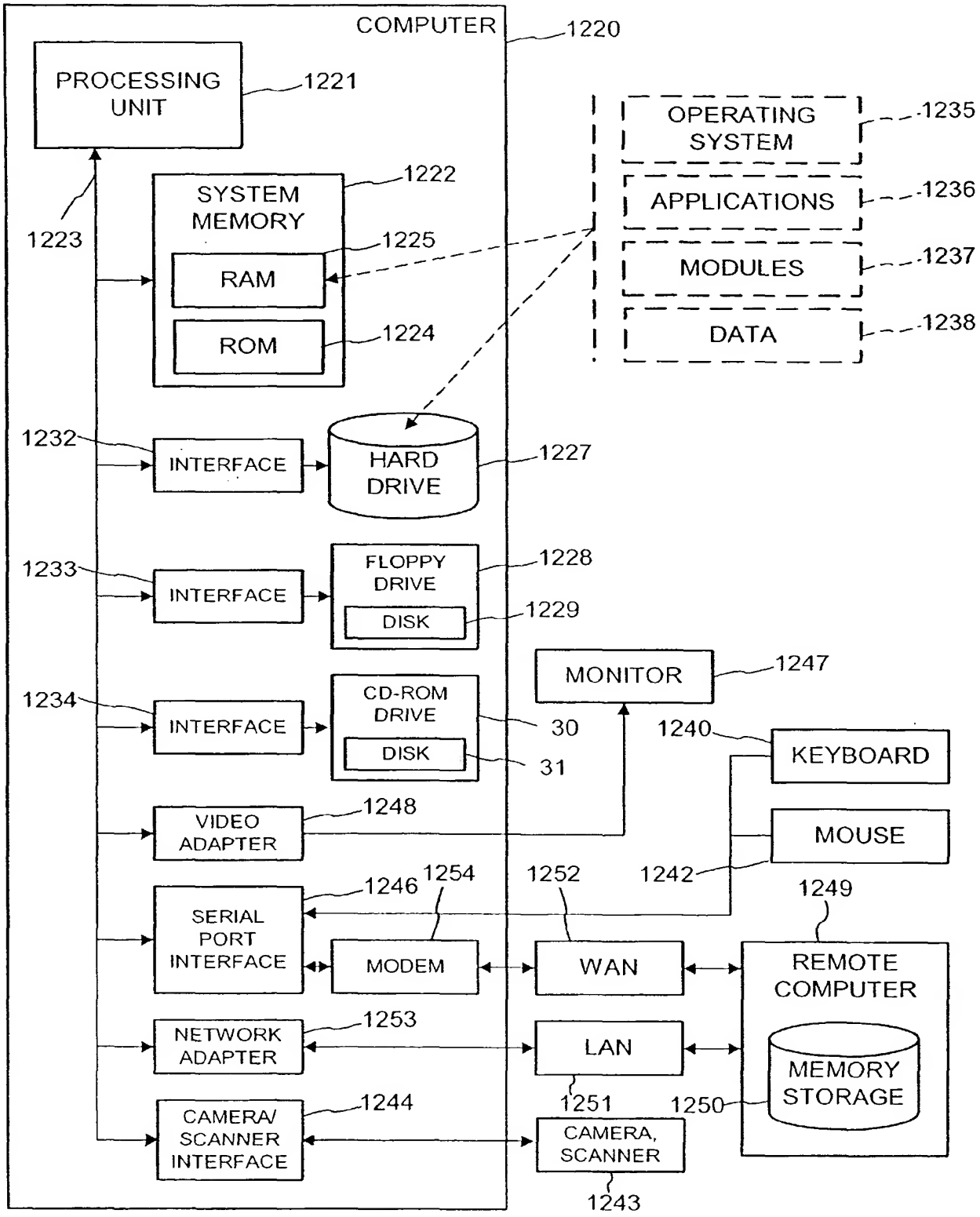

COMPUTER
1220
PROCESSING UNIT
1221
1223
SYSTEM MEMORY
1222
RAM
1225
ROM
1224
OPERATING SYSTEM
1235
APPLICATIONS
1236
MODULES
1237
DATA
1238
1232
INTERFACE
HARD DRIVE
1227
1233
INTERFACE
FLOPPY DRIVE
1228
DISK
1229
1234
INTERFACE
CD-ROM DRIVE
30
DISK
31
MONITOR
1247
1240
KEYBOARD
MOUSE
1242
VIDEO ADAPTER
1248
1246
1254
1252
1249
SERIAL PORT INTERFACE
MODEM
WAN
REMOTE COMPUTER
MEMORY STORAGE
1250
NETWORK ADAPTER
1253
LAN
1251
CAMERA/ SCANNER INTERFACE
1244
CAMERA, SCANNER
1243

Figure 1 — 8

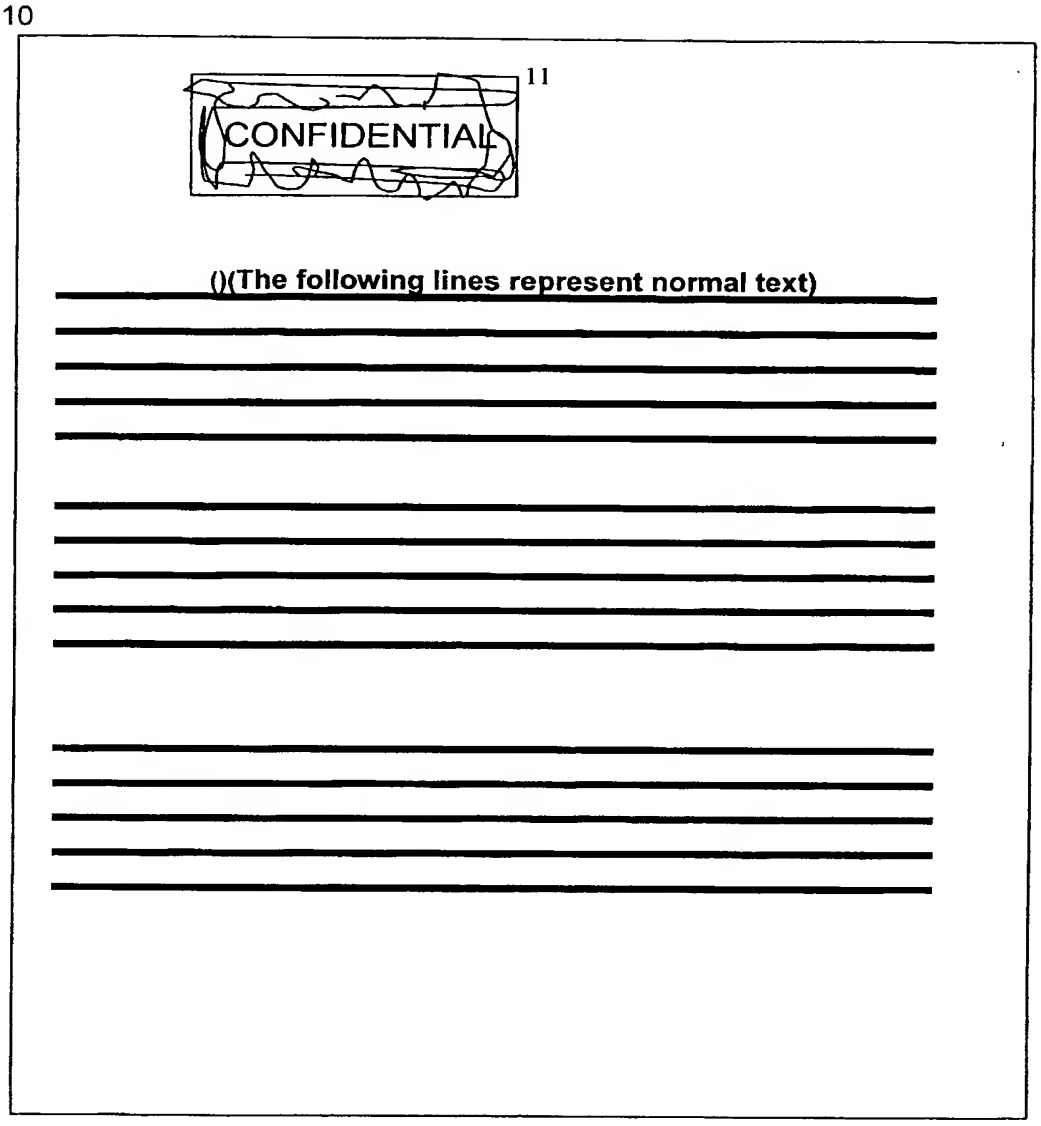

10
11
CONFIDENTIAL
()(The following lines represent normal text)

Figure 2 of 9

Fields in a typical watermark payload

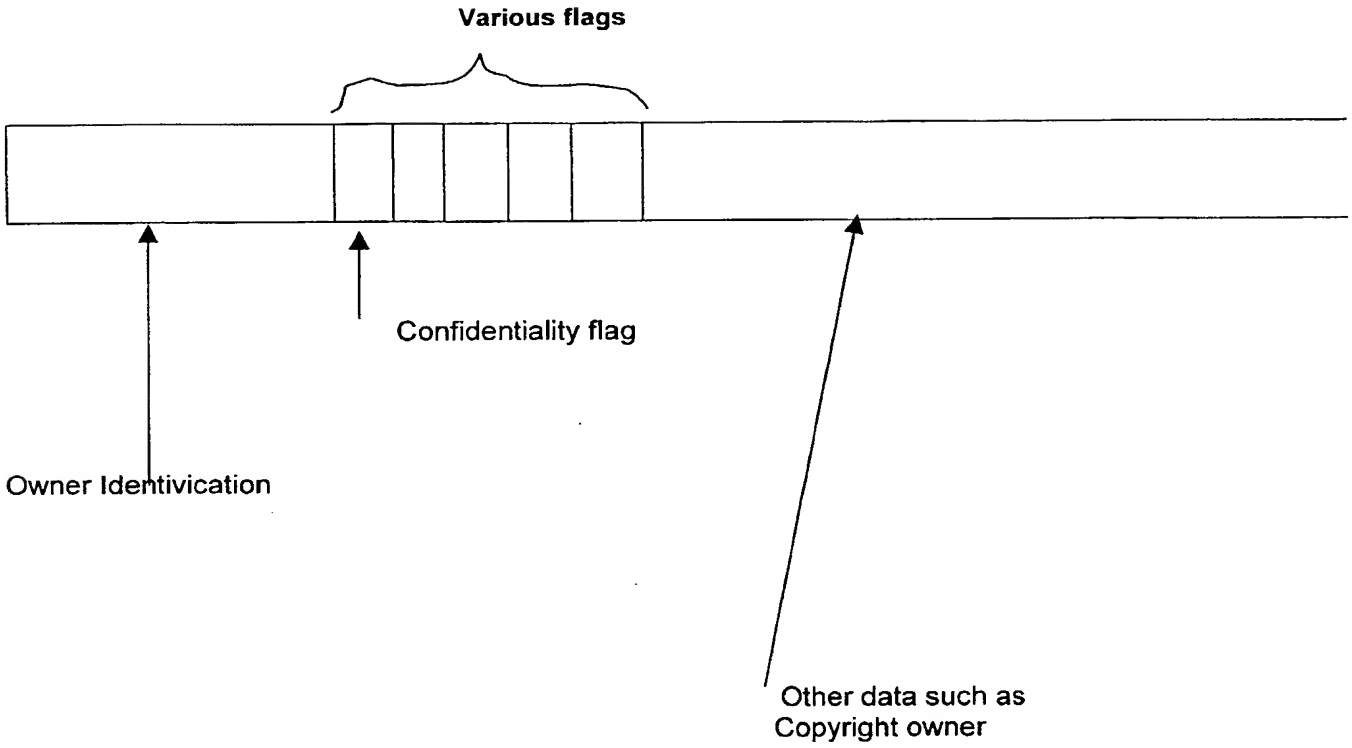


Figure 3 e-10

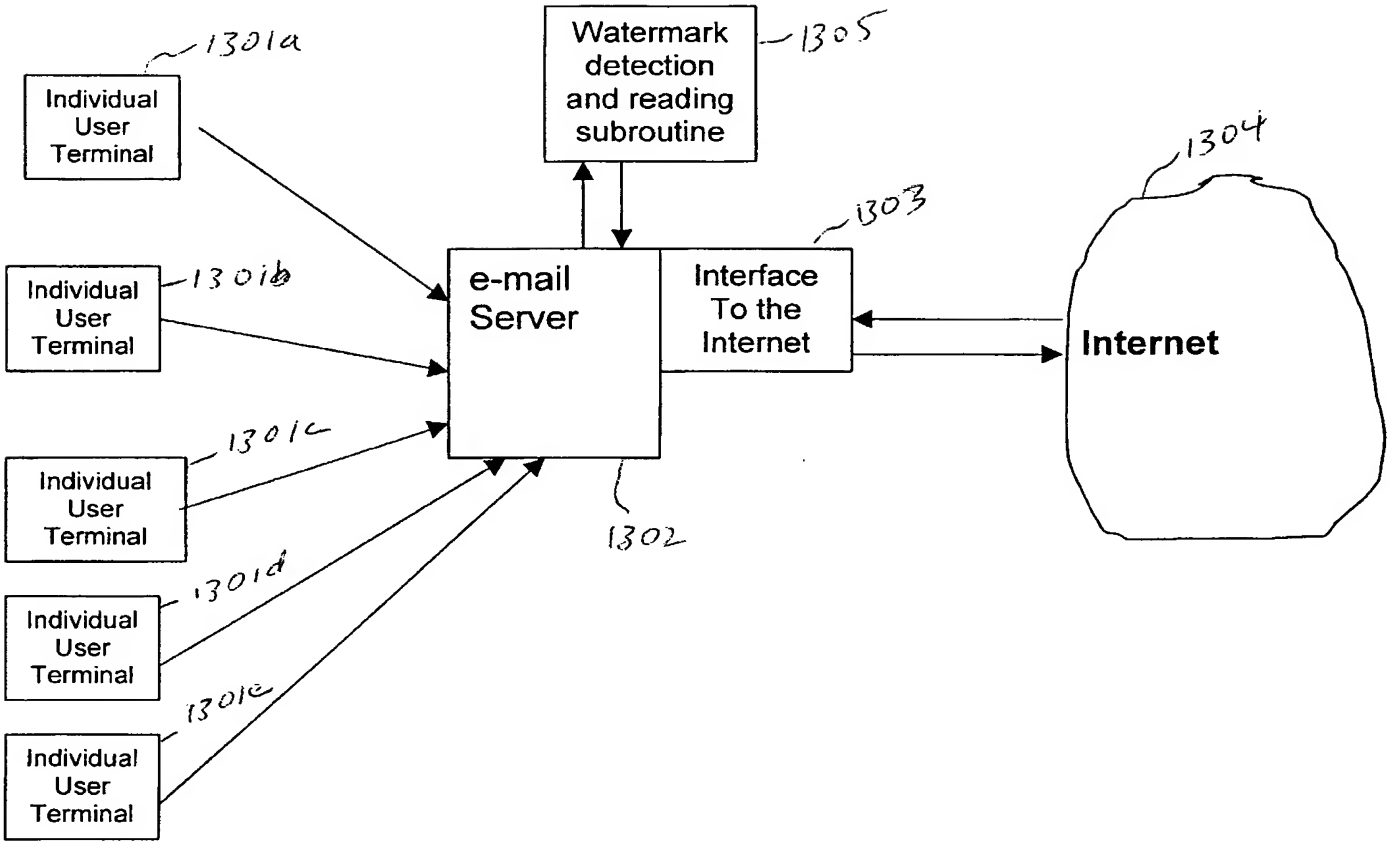

1301a
Individual User Terminal
Watermark detection and reading subroutine
1305
1304
1303
Individual User Terminal
1301b
e-mail Server
Interface To the Internet
Internet
1301c
Individual User Terminal
1302
1301d
Individual User Terminal
1301e
Individual User Terminal

Figure 4.

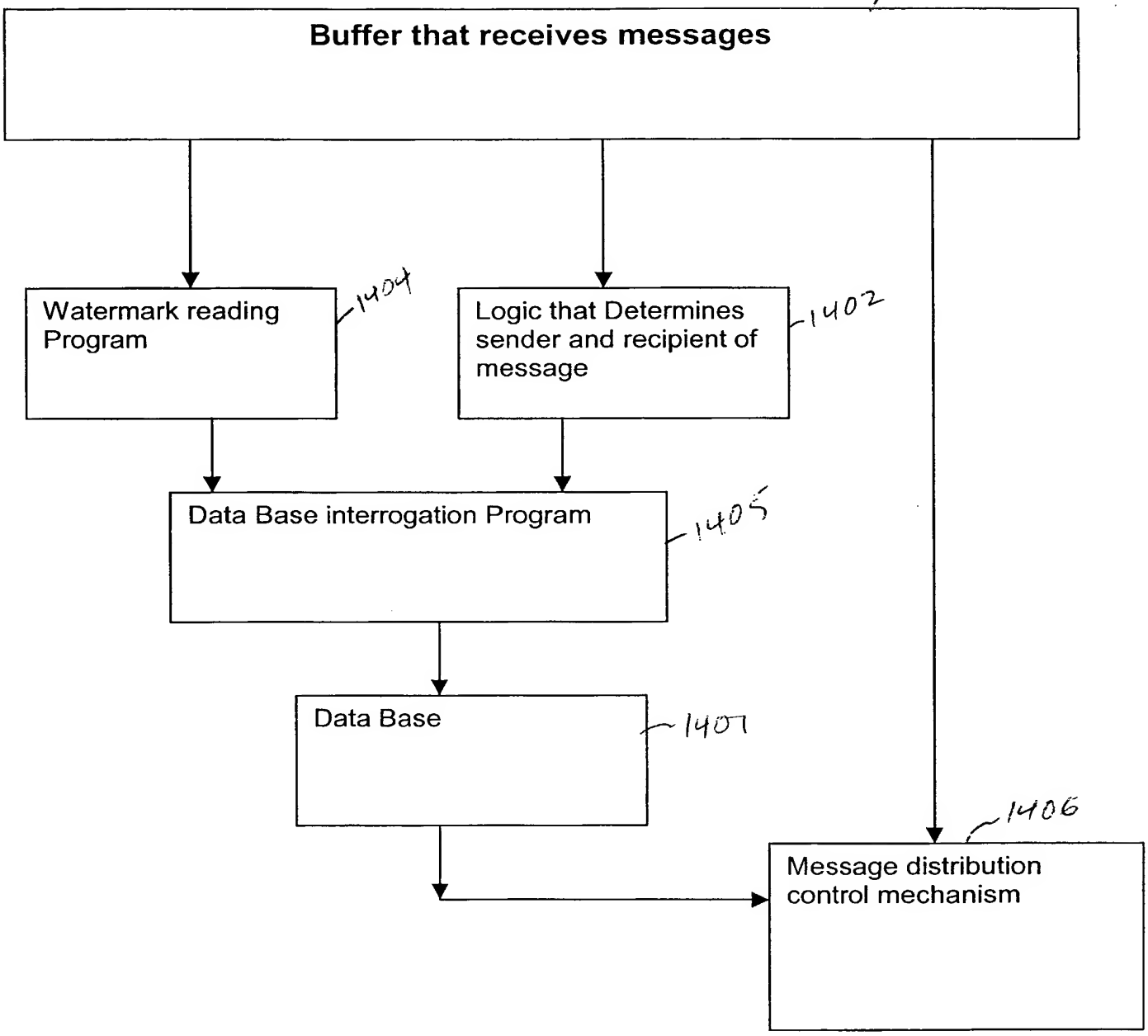
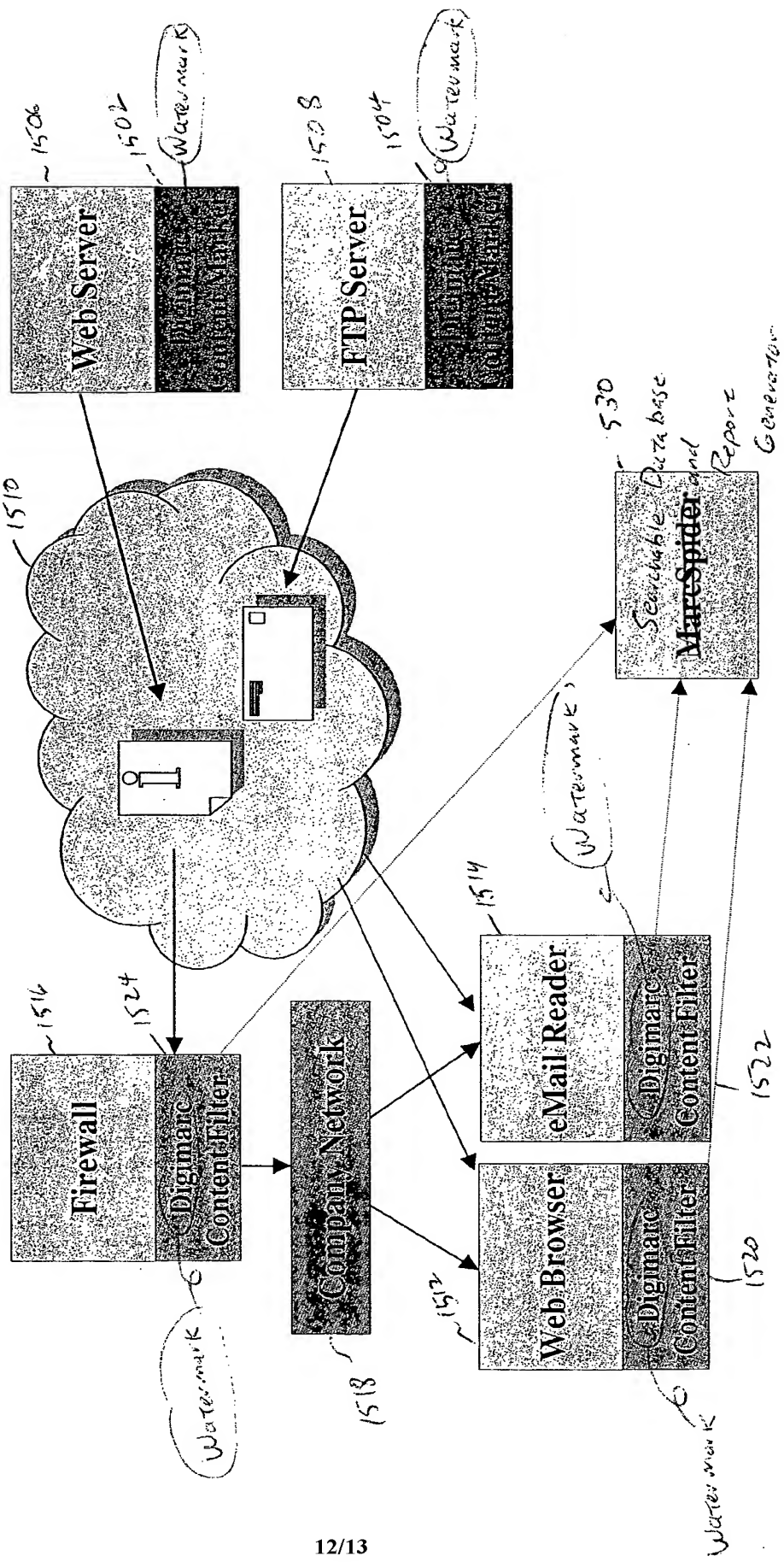

1403
Buffer that receives messages
Watermark reading Program
1404
Logic that Determines sender and recipient of message
1402
Data Base interrogation Program
1405
Data Base
1407
1406
Message distribution control mechanism

Fig. 12


Content Filtering System
Web Server
Digimarc Content Marker
Watermark
FTP Server
Digimarc Content Marker
Watermark
Firewall
Digimarc ContentFilter
Watermark
Company Network
Web Browser
Digimarc ContentFilter
eMail Reader
Digimarc Content Filter
Watermark
MarcSpider
Searchable Data base and Report Generator
1506
1502
1508
1504
1510
1516
1524
1518
1512
1520
1514
1522
1530

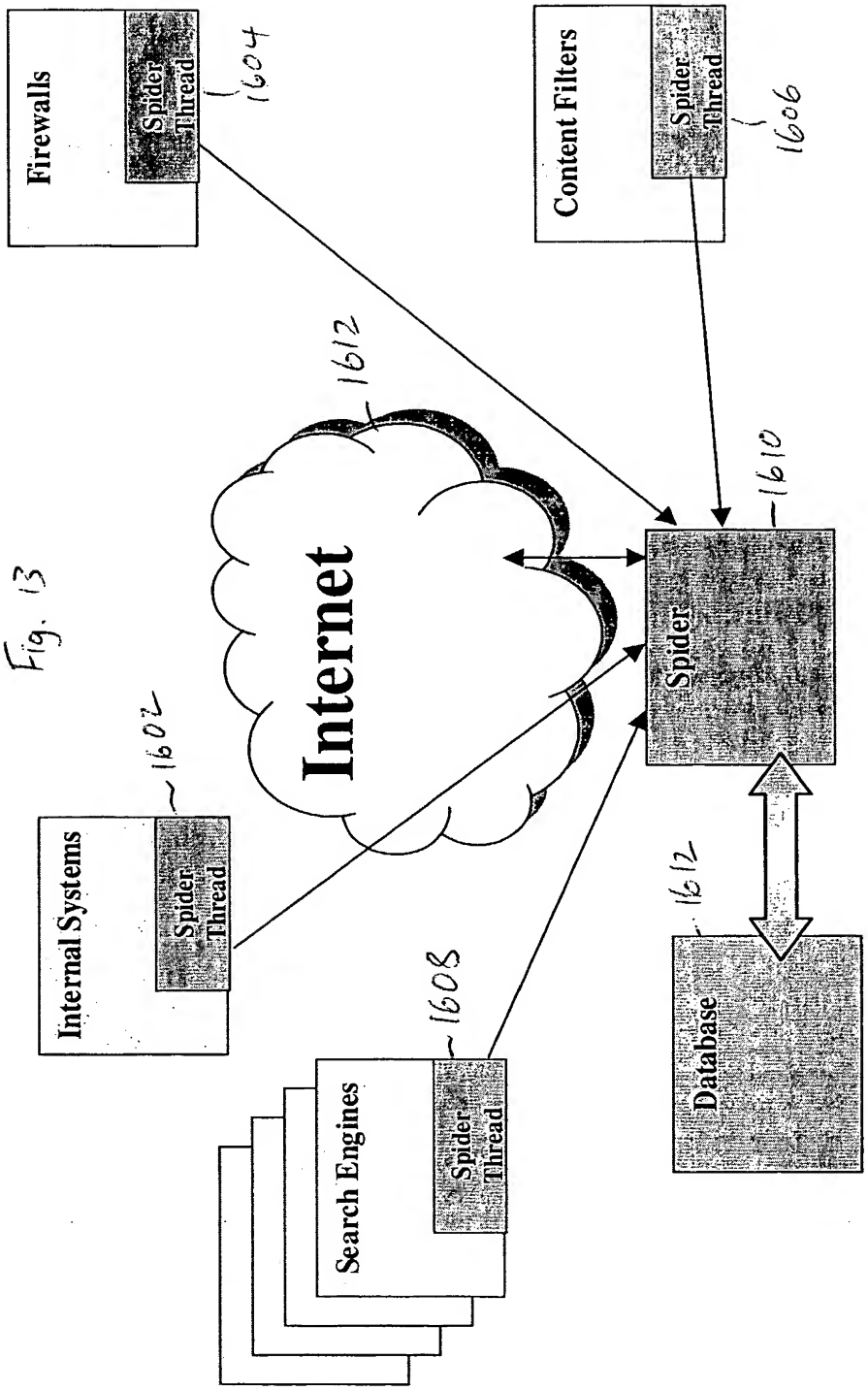

Firewalls
Spider Thread
1604
Content Filters
Spider Thread
1606
1612
Internet
1610
Spider
Internal Systems
Spider Thread
1602
1612
Database
Search Engines
Spider Thread
1608

Fig. 13

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/04812

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 13/00, 15/16; H04L 9/00

US CL : 709/217; 380/28; 345/335

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/217; 380/28, 54, 279; 345/335, 339; 283/72, 113; 713/161, 176

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,956,716 A (KENNER et al) 21 SEPTEMBER 1999, column 20, lines 22-column 27, lines 61, column 32, lines 38-column 33, lines 12, column 23, lines 25-49, column 32, lines 64-column 33, lines 12, column 23, lines 66-column 24, lines 14, column 27, lines 64, lines 64-column 28, lines 17, column 29, lines 27-35	1-20
Y	US 5,841,978 A (RHOADS) 24 NOVEMBER 1998, column 2, lines 9-26, column 58, lines 34-column 59, lines 47	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

01 June 2001 (01.06.2001)

Date of mailing of the international search report

28 JUN 2001

Name and mailing address of the ISA/US

Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

Tadesse Hailu

Telephone No. (703) 306-2799

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

tional application No.

PCT/US01/04812

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claim Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claim Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of Item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
GROUP 1: CLAIMS 1-20; GROUP II: CLAIMS 21-25; GROUP III: CLAIMS 26-32

The inventions listed as Groups I, II and III do not relate to a single inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical feature of Group I invention is the browser system claimed therein while the special technical feature of the Group II invention is the electronic massaging system and the feature of Group III invention is content filtering system. Since the special technical feature of the Group I invention is not present in Groups II and III inventions being claimed and the special technical feature of the Group II invention is not present in Groups I and III inventions being claimed and the special technical feature of the Group III invention is not present in Group I and II inventions being claimed, unity of invention is lacking.

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☒ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.: 1-20

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.